REPORT DOCUMENTATION PAGE

Form Approved OMB No. 074-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 30 Oct 1996	3. REPORT TYPE AND DATES COVERED Final

4. TITLE AND SUBTITLE Linkage of Constructive Virtual Simulations	5. FUNDING NUMBERS N61339-96-D-0002
6. AUTHOR(S)	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Lockheed Martin Corporation Information Systems Company 12506 Lake Underhill Road Orlando, FL 32825	8. PERFORMING ORGANIZATION REPORT NUMBER ADST-II-CDRL-022R-9600385
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) NAWCTSD/STRICOM 12350 Research Parkway Orlando , FL 32826-3224	10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
A - Approved for public release; distribution unlimited.

19970505 213

13. ABSTRACT *(Maximum 200 Words)*

1. Constructive simulations with aggregate level models simulate battlefield engagements at the level of military units (e.g., a tank company). Virtual simulations with entity level models simulate battlefield engagements at the level of individual platform or fireteam entities. Each entity level object is able to act independently and may be a representation generated by a man-in-the-loop (MITL) simulator or by computer-generated forces (CGF) software. Ideally, the outcome of an engagement should be the same whether the simulation is done in the constructive (C) domain, the virtual (V) domain, or a linked combination of the two (C+V). In practice, the results often differ; the difference is called combat results correlation error (CRCE). The broad scope of this research area is to investigate and devise methods by which CRCE can be classified, measured, and controlled. This document is divided into four sections as follows: The first section reviews the area of entity and aggregate simulation models and issues that arise when these models are linked, The next section derives the aggregate combat model, describes the calibration procedure for the aggregate model, and discusses the experiment and its analysis, The third section discusses conclusions, and The last section recommends future work in the area of entity and aggregate models and their linkage.

14. SUBJECT TERMS Simulation;Protocols;MODSAF;	15. NUMBER OF PAGES 48
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

DTIC QUALITY INSPECTED 1

ADST II-CDRL-022R-9600385
30 October 1996

ADVANCED DISTRIBUTED SIMULATION TECHNOLOGY II (ADST II)

CDRL AB01

LINKAGE OF CONSTRUCTIVE VIRTUAL SIMULATIONS

DO #008

FINAL REPORT

FOR: NAWCTSD/STRICOM
12350 Research Parkway
Orlando, FL 32826-3224
N61339-96-D-0002
DI-MISC-80711

BY: Lockheed Martin Corporation
Martin Marietta Technologies, Inc.
Information Systems Company
12506 Lake Underhill Road
Orlando, FL 32825

Approved for public release; distribution is limited.

Document Control Information

Revision	Revision History	Date
-	Original release	
-		
-		
-		
-		
-		
-		
-		
-		
-		
-		
-		
-		
-		

Table of Contents

1. Introduction

1.1 Scope

This report is submitted in fulfillment of Data Item Number AB01, Contract N61339-96-D-0002.

1.2 Statement of the Problem

Constructive simulations with aggregate level models simulate battlefield engagements at the level of military units (e.g., a tank company). Virtual simulations with entity level models simulate battlefield engagements at the level of individual platform or fireteam entities. Each entity level object is able to act independently and may be a representation generated by a man-in-the-loop (MITL) simulator or by computer-generated forces (CGF) software. Ideally, the outcome of an engagement should be the same whether the simulation is done in the constructive (C) domain, the virtual (V) domain, or a linked combination of the two (C+V). In practice, the results often differ; the difference is called combat results correlation error (CRCE). The broad scope of this research area is to investigate and devise methods by which CRCE can be classified, measured, and controlled.

CRCE is made up of various components. Among these components are:

- scenario correlation error (SCE) - changes in scheme of maneuver;
- terrain resolution correlation error (TRCE) - e.g., changes in intervisibility;
- performance correlation error (PCE)- e.g., changes in probabilities: P(kill/hit), P(detection)/time;
- behavioral correlation error (BCE) - e.g., assumed behavior in C vs. actual CGF behavior; and
- force resolution correlation error (FRCE) - e.g., different echelons of representation.

The LCVS delivery order investigated a single component of CRCE, FRCE. This approach strived to keep all other factors constant (e.g., terrain, scenario, etc.) with the resolution varying between ModSAF and Eagle vignettes as the V and C simulations respectively.

The method used in this first phase of the effort is to:

1. construct isomorphic vignettes for Eagle and ModSAF,
2. select aggregate attributes for measuring CRCE (FRCE),
3. verify compatibility of these measures,
4. add aggregate level objects to ModSAF, and
5. verify equivalence of force actions (including recalibration as required).

The aggregate attributes selected are attrition and engagement duration. Aggregate level objects are based on previous Corps Level CGF (CLCGF) work.

1.3 Document Overview

This document is divided into four sections as follows:

1. the first section reviews the area of entity and aggregate simulation models and issues that arise when these models are linked,

2. the next section derives the aggregate combat model, describes the calibration procedure for the aggregate model, and discusses the experiment and its analysis,
3. the third section discusses conclusions, and
4. the last section recommends future work in the area of entity and aggregate models and their linkage.

Four appendices follow:

1. the first one discusses the software requirements and design for LCVS,
2. the next one lists input parameters for LCVS executions of ModSAF (C),
3. the third one lists the input factor values and the output data from the simulation runs, and
4. the last contains three screen prints of ModSAF executions.

2. Background

As stated in the Feasibility Analysis Report (Report No. ADSTII-CDRL-A009-002, 1995), the principle benefits of a practical, credible linkage of aggregate and entity level simulation models are:

- providing a convenient way to model conflicts involving large numbers of platforms and weapon systems to support training of and operational analysis for higher command echelons at reasonable cost.
- allowing the introduction of a new system or concept into a large context environment during the concept development, system development, and testing phases of the procurement process.

Secondary benefits of a practical linkage of aggregate and entity simulation models include:

- allowing detailed inspection of specific portions of an engagement (by modeling them in the virtual domain)
- providing a broader context of conflict in which to conduct training and other activities with individual units and troops.

Tertiary technical benefits of a practical aggregate and entity simulation linkage include the possibility of using the best features of each class of model to remedy deficiencies in the other class.

The advent of virtual simulations with entity level models in the late 1970s and 1980s (e.g. USAF CASM; SIMNET OPFOR SAF and its later derivatives) seemed to provide a model of combat in which the causes of outcomes could be inspected at any desired level of detail to verify, validate, and accredit the model. Unfortunately, computer-generated semi-automated force (SAF) technology requires very substantial computer and communications power, even by present day standards, to simulate conflict in a causally consistent manner. While it is practical to model battles involving a few hundred platforms with a time resolution of a few seconds, it is very costly to expand such engagements to thousands of participants. The computational load of SAF algorithms grows roughly linearly with the number of simulated platforms. Some load components (intervisibility and engagement calculations) grow quadratically with relative force density (number of units within engagement range).

The current concept of SAF modeling (i.e. CCTT SAF and ModSAF) has evolved "bottom-up" from simpler systems concerned primarily with the visually realistic placement of individual OPFOR vehicles on three-dimensional terrain shared with networked man-in-the-loop (MITL) simulators. Throughout this evolution of SAF models, the higher level and larger scale direction of force behavior has been delegated to human operators working through a man-machine interface (MMI) designed for easy graphical control of an engagement. As a result, even the best SAF implementations do not provide detailed or validated models of the command and control structure and operation of the simulated force.

Scaling existing SAF systems to large engagements therefore requires ever-increasing numbers of both computers and human "command" operators. This has made it impractical routinely to apply entity level modeling techniques to simulation of conflict with large numbers of platform level participants. Although entity level simulations have been successfully demonstrated with a few thousand platforms (e.g. Zealous Pursuit, Zen Regard, and STOW-E), these demonstrations have been complex and costly to organize and execute. Ways must be found to provide command level training and operational analyses with less investment in exercise organization, preparation, and execution.

Existing constructive simulations with aggregate models (e.g., BBS, CBS, and Eagle) approximate the behavior of large aggregations of forces with much less computing resources than virtual simulations. To accomplish this, these aggregate models typically reduce the resolution of detail in the terrain, record only the central location of the aggregate forces, and adjudicate conflicts by use of Lanchester or other analytic models. In spite of the simplicity of their geographic and force-exchange models, they achieve high credibility by convincingly depicting the deployed force structures and the interactions occurring among different command echelons.

In a typical training application, aggregate models well describe the approximate movement, engagements, and command and control activities of command echelons directly interacting with the echelon of interest (normally one higher and two lower). For example, division level Eagle provides credible predictions of behavior and outcomes, including command and control activities, for brigade, battalion, and company echelons. But such systems provide no way to inspect or analyze the contribution of lower level units to the total outcome.

A new generation of simulation systems with aggregate level models is emerging (e.g. WARSIM 2000) that will use a new architecture (i.e. the JSIMS Architecture) to achieve the flexibility, extensibility, and adaptability to operational equipment provided by entity level models, with computation and communications resources only moderately greater than traditionally associated with constructive simulations. Such new-generation systems are being developed with aggregate and entity linkage in mind; however, the aspects of model design required to control or resolve CRCE in such linkages are poorly understood. Existing (legacy) constructive simulations do not offer ideal opportunities to resolve CRCE, since there will likely be significant changes required to the details of the simulation code, but these systems do offer the only currently accredited framework in which to examine CRCE realistically.

2.1 Variable-Resolution Modeling

The military operation research community has developed a substantial body of literature on the subject of variable resolution modeling during the past few years. Paul K. Davis and his associates at RAND have been noteworthy in working to further understanding of the very real problems of obtaining consistent variable-resolution models of the battlefield. Three recent papers, reporting work performed for DARPA and DMSO, are especially relevant to the current study.

In *An Introduction to Variable-Resolution Modeling and Cross-Resolution Model Connection* (RAND R-4242-DARPA), Dr. Davis provides a rigorous definition of model consistency and presents a theory of how models with variable-resolution capability ought to be constructed, which he calls integrated hierarchical variable-resolution modeling (IHVR). He illustrates how failure to design for variable-resolution can create problems. He argues that rapid prototyping and iterative development, as practiced in legacy DIS programs, are not a good way to arrive at workable models.

In *Cutting Some Trees to See the Forest: On Aggregation and Disaggregation in Combat Models* (RAND R-4250-DARPA), Richard J. Hillestad and Mario L. Juncosa explore aggregation and disaggregation in "square law" Lanchester combat models. They conclude that simply scored approaches that attempt to evaluate force components in isolation will not succeed. They show that partial aggregation and disaggregation of previously aggregated results is only possible when there is a constant mutual proportionality in the effects and vulnerabilities of the weapons of one side with respect to all of the weapons of the other side. Hillestad and Juncosa's results show it is impossible to resolve combat results inconsistencies within a common family of simple, aggregate level models that differ only in the aggregation measure of force "strength" and otherwise operate on under *mathematically identical assumptions of terrain, performance, and behavior.* This indicates that it is hopeless to expect to totally remove all CRCE from any useful variable resolution model, including the aggregate and entity linkages considered here.

In *Experiments in Variable-Resolution Combat Modeling* (RAND N-3631-DARPA), Richard J. Hillestad, John Owen, and Don Blumenthal examine the effects of varying model resolution on predicted combat results for very simple stochastic and deterministic simulations of two extremely simplified vignettes. They show that the combat results obtained are exquisitely sensitive to minute changes in such parameters as rate of advance, engagement range, weapon effectiveness, firing rate, virtual unit size, and simulation time step. Although the examples used are trivial, and the various strength and mobility factors were chosen to emphasize the domain in which the combat results behave chaotically, it can be argued that such effects are likely to occur in a wide range of more typical engagements. For example, a minor deflection in terrain elevation, accounted for in one model and not accounted for in another, can often induce enough difference in mobility or engagement range or both to reverse the outcome of an otherwise highly detailed entity level simulation.

2.2 Aggregate/Entity Level Modeling

2.2.1 Aggregation/Disaggregation Concepts

The concept of aggregation is to represent a collection of entities by simulating a single object that maintains information about that group and, when necessary, provides information for disaggregating itself into the individual entities. Individual physical characteristics, such as positions, speeds, fuel levels, damage levels and entity behavior, such as battle damage and attrition, are predicted by the use of deterministic and statistical algorithms for the aggregate unit.

Due to a variety of reasons, an aggregate unit may disaggregate. Contact with a manned simulator or interaction with another simulated object that requires higher resolution methods to be resolved are some of the reasons for disaggregating. Coordination and synchronization of the aggregate unit with its corresponding enitities must take place in order for the simulation to remain valid. One way is to pass control of the information to the disaggregated entities by allowing updates only at the entity level. Messages are sent to aggregate unit echoing the change if the aggregate unit co-exists with the disaggregated entities. Another method is to reequilibrate at the aggregate level when the entities are reaggregated back into an aggregate unit (Davis, 1995). Reequilibrate is a general concept which can include alertness, allocation of fires, redeployment of command and control assets, and maneuver of gound forces, aircraft, and ships.

There are various levels of disaggregation that can occur: partial, psuedo, deaggregation. Partial disaggregation is when only part of an aggregate unit is created as entities. For example, only entities in contact with opposing forces need to disaggregate from the aggregate unit for combat. Pseudo disaggregation is when entities are created from the aggregate unit but control over entity data is maintained by the aggregate unit. This allows other entities to make contact with the disaggregated entities without committing the computer resources required by fully simulated entities. Deaggregation is when the entities are created and the aggregate unit ceases.

The aims of including aggregation as a simulation modeling technique are to save computer resources, thereby enabling large-scale training scenarios, and for simulations that do not execute in real time, a reduction of the simulation execution time (Kester, 1996). The use of disaggregation impacts these aims and hence multi-resolution systems try to limit its use through careful design.

2.2.2 Legacy and Contemporary Systems

One current methodology for achieving variable resolution modeling is to link existing systems developed for different levels of resolution. Simulation systems are broadly classified as either aggregate level, also known as constructive, simulations or as entity level, also known as virtual simulations. This classification will no longer be valid as new systems which are incorporating both aggregate and entity level modeling capabilities come online.

Constructive+Virtual linkage (Stober, Kraus, Foss, Franceschini, and Petty, 1995), or more recently, Aggregate+Virtual linkage (Schricker, Franceschini, Stober, and Nida, 1996), is the term applied to battle field simulations that link systems developed at different levels of resolution. The common approach for handling interactions between objects in the constructive world with objects in the virtual world has been to disaggregate/reaggregate the constructive objects and use models and approaches in the virtual world for the interaction. Disaggregation can be specified to occur in designated areas of the battlefield known as virtual playboxes (Karr, 1994) or can occur dynamically when specified conditions are met. For example, disaggregation can occur when an entity makes contact with an aggregate unit of the opposing force.

Technical challenges need to be addressed for aggregate+virtual linkages to be successful (Peacock, Bombardier, Panagos, 1996). Examples of these are:

1. extension of DIS to include aggregate level protocols,
2. modification of constructive simulations to incorporation of DIS entity level information,
3. development of a dynamic aggregation/disaggregation protocol,
4. interaction of objects in the aggregate+virtual world be consistent and valid regardless of whether the interaction takes place under the control of the aggregate or the entity environment.

There are successful linkages of battlefield simulations. Examples are: Eagle to BDS-D, Corps Level Computer Generated Forces (CLCGF), AWSIM Interoperability with ModSAF (AIM), and Janus Combat Model Link to DIS (Jlink). Each linked system has developed an interface architecture that links the aggregate units to the virtual entities that exist in a DIS world.

2.2.2.1 Eagle/BDS-D Linkage

The Eagle/BDS-D linkage (Schricker, Franceschini, Stober, Nida, 1996; Franceschini and Clark, 1994; Karr and Root, 1994) is a system that links Eagle with a DIS/SIMNET virtual environment using the IST CGF testbed. Eagle is a corps and above level constructive combat simulation. This system deals with issues as: aggregation and disaggregation of units, synchronization of the constructive time-stepping with the real-time virtual clock, conversion of terrain coordinates, representation of aggregate units on the entity level terrain.

Eagle models the aggregate level units. An aggregate unit is controlled by Eagle unless it moves into a prespecified high fidelity area, in which case it disaggregates. If a unit disaggregate, an operations order is created and executed by the CGF testbed. A vehicle placement algorithm (Franceschini and Clark, 1994) is used to place the entities on the terrain. Information about the disaggregated unit is compiled and sent to Eagle who is no long simulating that object. When the re-aggregation occurs for a unit, the Eagle simulation starts simulating that unit again and the entity level simulation receives aggregate data regarding the unit.

This linked system has an aggregate protocol, the Interoperability Protocol (IOP), for transferring information detailing aggregate and entity level status across the A+V boundary. Five main message types (Schricker, Franceschini, Stober, Nida, 1996) are:

1. initialization messages establish communications pathways when a simulated object joins the exercise,

2. unit status messages allow the exchange of aggregate and entity level information,

3. state transition messages allow units to request aggregation or disaggregation,

4. operational detail messages provide a means of communicating a unit's operations and intentions, and

5. indirect-fire messages allow aggregated and disaggregated units to exchange indirect fire.

Eagle/BDS-D project is focusing on three areas that make A+V linkage difficult. These are: IOP message size, direct fire between aggregate units and virtual entities, and spreading disaggregation. An IOP message can grow in size since it can carry data concerning hundreds of individual virtual entities. A quick solution to the problem of direct fire among multi-resolution objects is to disaggregate all the objects involved. This leads to the third problem of spreading disaggregation. One disaggregation can trigger others and may lead to an overload of resources in the virtual environment. This reduces the scalability of the system which defeats one of the objectives for linking an aggregate and with a entity level simulation.

2.2.2.2 *Corps Level Computer Generated Forces (CLCGF)*

Joint Precision Strike Demonstration (JPSD) program has sponsored the construction of the Corps Level Computer Generated Forces (CLCGF) (Peacock, Bombardier, Panagos, 1996). CLCGF system integrates the constructive, aggregate level simulation system Eagle with the virtual, entity level simulation system, ModSAF. It is responsible for simulating the entities and aggregate units on the corps battlefield, managing a plan view display, interacting with other DIS simulations, and interfacing with the other non-DIS systems.

Currently resolution change in CLCGF is controlled by three rules:

1. unit-in-area rule: if an aggregate unit enters a specified area, it disaggregates
2. duration: if an aggregate unit has been disaggregated for longer than a specified time it re-aggregates,
3. sphere-of-influence: if another type of vehicle comes within a radius to the aggregate unit, then disaggregate.

The second rule happens automatically unless a disaggregate request message is sent periodically. The last rule is not currently implemented (Peacock, Bombardier, and Panagos, 1996).

Whenever an aggregate unit controlled by Eagle is ordered to disaggregate Eagle relinquishes control of the aggregate. ModSAF assumes responsibility for simulating the entities created by the disaggregation. ModSAF unit is given an operations order based on its mission in Eagle. Entity state information is summarized by ModSAF and sent to Eagle. Eagle maintains the location and compositional data for the disaggregated unit.

The objective of CLCGF is to provide the corps level simulation environment for DIS exercises. CLCGF is used to simulate maneuver and artillery units contained in an Army corps. Transmission of unit state data at the aggregate level significantly reduces the number of PDUs transmitted and thereby reduces the network load. If the receiving systems need the aggregate unit information in terms of the individual entities it can pseudo-disaggregate by translating the information into entity level information. In this way the DIS network is not flooded.

2.2.2.3 *Janus LINKed to DIS (JLINK)*

Janus Combat Model (Pratt and Johnson, 1995) is a constructive simulation used by the U.S. Army. Janus has two objectives. First it is testbed for studying weapon systems and tactics. Second it is a training tool for teaching tactics and staff planning. Janus can also train unit staffs in Command and Control.

Janus models individual entities and aggregate level units. Scenarios deal with force-on-force combat between two opposing forces. Many of the models used by Janus have been through a rigorous validation and verification process. Janus however is a deterministic, discrete, closed, and event-driven system. The JLINK project is an attempt to link Janus to BDS-D, a real time virtual DIS system.

This aggregate+virtual linkage is not a linkage of an aggregate simulation system with a virtual system since Janus simulates both levels of resolution. Instead the linkage is an interface to DIS. Accomplishing this interface with DIS involved many changes to Janus' algorithms to make it DIS compatible. Analytic algorithms, such as, night detection, damage probabilities, aircraft play, dead reckoning, and engagement arbitration, were modified. Filters had to be developed to extract data to and from DIS.

Janus as a self-contained simulation model of combat performs the fire, impact, and kill arbitration for all its entities. Linkage with DIS changed this. Entities under DIS control can only be damaged if the entity itself determines that it is due to a weapon impact. Janus maintains information about DIS entities as a local ghost. Entity state PDUs update this information.

The JLINK project merges with DIS virtual entities. The DIS environment has added man-in-the-loop capability and reactive behavior to Janus. JLINK goal is to provide a validated simulated force for armor scenarios. JLINK does not interact with DIS at the aggregate unit level.

2.2.3 Communication Protocols

The simulation industry has been using different methods to achieve aggregate level communications. The following paragraphs are brief descriptions of some of these methods :

2.2.3.1 Aggregate Level Simulation Protocol (ALSP)

The ALSP is a research and development project responding to a desire to be able to re-use known reliable Service models to train in a Joint environment It is a generic protocol designed to interoperate aggregate, time-manager simulations, and Joint Training Confederation (JTC). The ALSP is based on four design principle from SIMNET:

1. distributed computation based on combat entity ownership;
2. avoidance of single critical resources;
3. reliance on broadcast communications; and
4. replication of a limited set of combat entity attributes among all simulators.

The ALSP consists of two peer level protocols and a vertical connection protocol. The upper protocol layer carries information concerning the interactions of the battlefield entities. The lower protocol layer provides simulation time regulation and message transportation service.

JTC is the primary application for ALSP. The 1996 Joint Training Confederation consists of eight simulations interoperating through the ALSP Infrastructure Software (AIS). These eight simulations are: the Air Warfare Simulation (AWSIM), the Research, Evaluation, and Systems Analysis (RESA) model, the MAGTF Tactical Warefare Simulation (MTWS), the Corps Battle Simulation (CBS), the Tactical Simulation Model (TACSIM), and the Combat Service Support Training Simulation System (CSSTSS).

2.2.3.2 High Level Architecture (HLA)

HLA is an attempt to capture the best ideas derived from experience with both ALSP and DIS, but with increased potential for interoperability of different simulations and reuse simulation components. The HLA is intended to support the entire range of modeling and simulation activities which includes :

a) real-time, man-in-the-loop simulation which are conducted using DIS protocols;
b) aggregate level constructive simulation that have been incorporated in the ALSP confederation;
c) hardware-in-the loop testing; and
d) operations analysis simulations.

Each HLA compliant simulation defines an explicit object model that can be used as the basis of negotiation within the federation to from a Federated Object Model (FOM). Aggregate information can be defined in the object model. The FOM is used to develop, initialize, execute and analyze the results of the interacting simulations. A common Run-Time Infrastructure (RTI) is used to mediate the interactions of the simulation within a Federation Execution.

The HLA can be described as a set of rules. These rules refer to four areas :

a) all simulations and federations describes their capabilities in terms of their object model;

b) simulations be developed with basic capability to interact with other simulations in specific way;

c) specifies federation wide services to support federations' runtime interactions; and

d) specifies the interface between the simulations and the RTI services.

In reference to the Fall 1996 DMSO newsletter, Volume 1 Number 3, HLA has been designated as the standard technical architecture for all DoD simulations. There will be no investments in non-HLA simulations after Oct 1, 99 and no non-HLA simulations will be used after Oct 1, 2001. HLA supersedes all past standards including DIS and ALSP. The memorandum was signed on September 10, 1996 by Dr. Paul Kaminski, the Under Secretary of Defense for Acquisition and Technology.

The HLA baseline is defined by version 1.0 of the HLA Rules, HLA Interface Specification, and the HLA Object Model Template. An "early" version of the RTI will be released in December 1996 and a follow-on version 1.0 will be released in late Spring 1997. This version of RTI will be released first as C++ executable code for Sun/Solaris machines, and later on to other platforms and languages.

2.2.3.3 DIS ++

The objective of DIS++ is to expand the standards development activities to include both the HLA and other members of the modeling and simulation community. DIS ++ is a set of standards supporting the HLA.. DIS++ is the next generation of DIS.

2.2.3.4 Command and Control Simulation Interface (CCSIL)

CCSIL is a special language for communicating between and among command entities and small units of virtual platforms generated by computers for the DIS environment. CCSIL includes a set of messages and a vocabulary of military terms to fill out the messages. The Communication Module of the Command Forces (CFOR) infrastructure services software allows the developer to package and unpackage the CCSIL contents in and out of the DIS Signal PDU.

The most important requirement for CCSIL is that the messages be interpretable by software with no human operators assistant. The current state of natural language interpretation software is not sufficient to support this requirement. CCSIL Release 1.1 consists of 37 messages types. These messages are grouped into 7 categories : Orders and Directives, Situation and Status Reports, Intelligence, Fire Support, Engineer, Air Defense and Combat Service Support.

2.2.3.5 Aggregate Protocol for DIS

The Aggregate Protocol provides a method for grouping multiple entities and communicating information about these group of entities. The main objective of the Aggregate Protocol is to save the number of Entity State PDUs being sent out on the network. This is achieved by some newly defined DIS PDUs such as Aggregate Descriptor PDU, De-Aggregate Request PDU, Re-Aggregate Request PDU, etc. A draft version of the Aggregate Protocol was completed; however, this effort has been put on hold due to the emerging HLA.

2.2.3.6 Persistent Object (PO) and Persistent Object Protocol (POP) for ModSAF

By interacting through the PO database, commands are distributed from any SAFstation to any SAFsim for execution. The PO protocol shares a large amount of information efficiently, provides real-time performance, and allows changes to the state of the objects. This is accomplished by periodically broadcasting Protocol Data Units (PDUs) which can describe an object, delete an object, request an object, etc.

2.2.3.7 SAF Entity Object Database (SEOD) and CGF Protocol for CCTT SAF

CCTT SAF used the basic concept in PO to develop SEOD and CGF Protocol to communicate CGF command and control information. SEOD is a distributed database which allows CCTT applications to store, retrieve, modify, delete and share CGF command and control information. These command and control information supplements the information provided by the DIS standard.

2.2.4 LCVS Extentions of DIS for Aggregate Model

Until now, the main focus of running Aggregate (Constructive) units in DIS, or virtual simulations, has been to get the aggregate units to play in the simulation using existing methods of interaction. Any focus on interacting Aggregates with other units without disaggregating them has been limited. Units have been displayed and moved around, but have mainly used disaggregation in order to interact. There is no existing protocol support to allow two constructive units to interact with each other at the constructive level. The closest there has been has been to have simple entity/aggregate interactions. This implementation involves shell fire being noted by the constructive simulations, and their determination of how each impact affects the aggregated unit as a whole.

Entity combat interactions are events, in that they happen and are complete. They maintain no state, and once issued, are finished. They do not depend on knowledge of the other entity's state (Technically you can do a combat interaction with a dead vehicle), and all that is required is knowledge of the other entity's position to calculate munition delivery. The entity delivering the munition is only required to have knowledge enough to deliver the weapon, and to notify the target that it has been hit with a particular munition at a particular location (or, in the instance of indirect fire weapons, where the munition impacted on the terrain). It is the job of the target to determine the damage.

Aggregate interactions are continuous. They start when the units engage, and end when units either disengage, or one becomes completely attrited. The Lanchester equations require the knowledge of the current state of the other unit that is being engaged to calculate it's own strength reduction. If implemented simply, the unit would have to continually receive the state of each other unit it was engaged with. Aside from being impractical in a networked simulation, it is probably impossible to do in a simulation where time is continually flowing forward . By the time the information is received across the network connection, it is no longer true since a small amount of time has passed since it was sent.

This is similar to the situation with entity location in current simulations, and begs for a similar solution. With entity movement in DIS, for example, the problem of knowing where an entity is located is solved by the concept of RVA (Remote Vehicle Approximation). Instead of trying to continually update the position of an entity, the standard is for an entity to publish three pieces of information:

1. the vehicle's current momentary state, in this case position,
2. an equation by which the entity's location in the future can be fairly well estimated from this base, and
3. the parameters to this equation.

As an example, the simplest form of RVA gives speed and direction of travel as parameters to the equation, which calculates the resulting linear velocity, and based on the time since the last time the position was reported, calculates where the vehicle should be now along the resulting vector. There is a published set of equations which are acceptable to use with the DIS protocol, though only a few are actually in widespread use.

This concept could readily be applied to aggregate interactions. For purposes of this discussion, these process will be referred to as Remote Unit Strength Approximations (RUSA). RUSAs would required that a standard set of equations be established that describe the various ways to estimate the rate of loss of unit strength, and the parameters to those equations. As with entity movement and RVAs, it is NOT a requirement that these equations reflect how unit strength is actually calculated in the particular aggregate simulation, only that the equations do a good enough job of estimating this change so as to be close enough that there is not a need to continually update a particular unit's strength value on the network. Each unit would be responsible for publishing the same three items that are present with entity RVAs:

1. the current state of strength,

2. which of the standard set of equations to use, and
3. the parameters for the equation.

As an example, if the current simple ModSAF aggregate model that was used for this project were used, the resulting RUSA implementation would probably be something like the following:

- Strength: A floating point number defined as a number of entities of a certain type (in the case of this experiment, only one type was used, tanks, which significantly simplified the interactions). Strength is reduced by fractional amounts over time until the unit is fully attrited.
- Equation to use: The Lanchester equation used here was fairly straight forward, and as such could probably be used directly. Attrition would be calculated using the same methods used in this experiment. Parameters for the above interactions.

A benefit of this approach is that it's implementation would better allow the possibility of interactions between existing Aggregate simulations. This assumes they could be modified to run at a common rate of time flow, most practically real-time, to keep pace with each other and the other non-aggregate simulations being run in the exercise. Both Eagle and BBS accomplish this currently using a methodology that involves advancing the clock by large fixed time steps, and then waiting for real time to catch up. They would also have to be modified to use an internal database that presents these simulations with a representation they could understand for the external units they would interact with. This database would have to maintain the state of these external entities, doing the RUSAs, for these external unit's representations.

One minor repercussion of a networked implementation of aggregate simulations that interact with entity simulations in real-time (ModSAF, etc.) would be that the results these simulations arrived at would become slightly non-deterministic. Due to timing issues between simulations on different hosts, calculations based on information received from other hosts and the receiving of updates of that information would at times be done in different orders, when compared between subsequent runs of the same simulation scenario. This would result in slight differences between the resulting calculations. However, these results would still be almost identical, since a single interaction would be spread out over a large number of calculations. Therefore, any localized differences in particular interactions at any one moment, when compared between two runs of a simulation, would mostly disappear in the average results. A similar situation exists with the stand alone aggregate implementation created for this experiment. Since the calculations were all based on time since last calculation, and since the simulation does not do fixed time steps, the point at which particular calculations are done varies between runs of the same scenario. As an example, to confirm that this was not a significant problem, there were a few runs done (five) of the same scenario with the ModSAF Aggregate implementation. The total number of vehicles lost in the scenario only varied by 0.10 amongst the runs, which when measured against the total losses in the scenario came out to a variation of plus or minus 0.18%, or less than 2/1000ths. Therefore this is not seen as being a significant problem.

On note needs to be made here. If RUSA calculations of the estimated value of the unit strength are to be consistent with each other on local and remote hosts, then the equations have to be defined in such a way as to be consistent in all locations regardless of tick methodology, which means structuring the calculations to be dependent on the start value and the change in time, instead of on the last arbitrarily timed estimate's (RUSA) resulting value. Otherwise, the local calculation, which is used to estimate how much the remote host's estimations of the Unit strength are off from the real value, will be different from those remote values, and therefore would be useless.

2.2.5 Spreading Disaggregation (SDA)

2.2.5.1 Current SDA Issues

This study did not include SDA in the experimental phase, since the experimental phase researched the calibration of a pure aggregate level combat with a pure virtual level combat. Disaggregation occurs when control of an aggregate unit is passed to the virtual simulation. The disaggregation transforms information about the unit into information about individual entities. The virtual system fills in any missing information since more information is needed than can be provided by the aggregate simulation. For example, an aggregate simulation generally does not store the terrain representation with as much detail as a virtual simulation does. An aggregate simulation can disaggregate a unit and approximately place its vehicles on the terrain in the correct formation, but it will not be able to place them in positions that meets the line of sight requirements without the virtual simulation providing the routines. For reaggregation, the reverse occurs.

Aggregation and disaggregation decisions are made when conditions are met. Not all aggregate+virtual linkage systems implement the same set of conditions for aggregation/disaggregation. One common approach to disaggregation is to disaggregate units that move into a virtual playbox (Karr and Root, 1994). Any units moving outside the playbox are reaggregated. This approach is simple but restrictive because the playboxes are not dynamically determined. Their boundaries are static and objects may cross the boundary sooner than is desirable.

A virtual playbox controls SDA by disaggregation only witin a defined high resolution area of the battlefield. This follows an approach previously explored by Mitre for applying Aggregate Level Simulation Protocol (ALSP) techniques to distributed linkage of aggregate level simulations (Dahmann, *et al.*, 1993).

In CLCGF, dissaggregation management is not limited to defined areas, but is based on an algorithm computing expected capbility to interact. This approach is to disaggregate a unit whenever it is within sensor range of a virtual entity. The drawback to applying this approach without constrainsts is that a single disaggregation may trigger a chain of disaggregations that ultimately overloads the DIS network. Consider also the case of aircraft virtual entity that flies over aggregate entities repeatedly. The aggregate entities disaggregate and reaggregate with every fly by and flood the DIS network and consume valuable CPU time. This drastically reduces the number of entities that the linked simulation system can handle.

Disaggregation whenever objects make contact is a method for reducing CRCE. Interactions among simulated objects are always handled at the entity level where the most detail is available. However, this technique can severely limit simulated force size and hence is not a viable solution. The new generation of training simulations, such as WARSIM under the JSIMS Architecture and Synthetic Theater of War (STOW), must meet the requirement of running exercises that encompass 100,000 entities with computation and communications resources that are only moderately greater than traditional constructive simulation systems. Therefore, SDA must be controlled without overly increasing CRCE.

2.2.5.2 Controlling SDA with Hybrid State Simulation

The Hybrid State Simulation (HYSIM) study's goal is to control SDA by the use of a hybrid aggregate+virtual architecture. This approach allows precise control of SDA for the measurement of CRCE effects.

A key concept of HYSIM is that virtual level entities do not interact directly with aggregate level units. When an aggregate unit, say X, comes in contact with virtual entities, the aggregate unit X is disaggregated and a pseudo aggregate unit, say Y, is created from the virtual entities. The pseudo aggregate unit Y is a bookkeeping device created for the aggregate unit X. Its role is to enable the aggregate unit X to compute the proper constructive statistics in order to guide the disaggregated entities created from X in their contact with the virtual entities that pseudo aggregate unit Y represents. As the combat exchanges are made, changes to the Y's virtual entities' states are transmitted to their associated pseudo aggregate unit Y. No information about the pseudo aggregate unit Y is broadcast or used by any object than the aggregate unit X.

Similarly, pseudo entities Yi are created from the aggregate unit Y as booking devices for the virtual entities Xi. The Yi entities do not have any simulated interactions with entities other than the Xi entities. As the aggregated unit Y combats other aggregate units, changes to Y are transmitted to the pseudo entities Yi. No broadcasting of this information is done.

When disaggregated entities Xi move, shoot, and take damage, their status is rolled up to modify the status of their associated aggregate unit X. The disaggregated entities Xi can undergo the usual virtual interactions. These interactions are constrained to produce the constructive results as predicted by the interaction of the aggregate unit X with the pseudo aggregates that represent the virtual entities the Xi are in contact with. Because the pseudo aggregate is a valid summary description of the virtual entities, this will always be possible if the constructive algorithms are representative of virtual interactions.

In this architecture, disaggregation spreads only one unit past the virtual entities and SDA is minimized. The pseudo aggregate units and pseudo virtual entities act as buffers to eliminate edge effects so that real aggregate units and real virtual entities do not perform discontinuous or unphysical behaviors.

2.2.5.3 Approaches without Disaggregation

Two other approaches that impact the SDA problem are worth noting. Both of these approaches attack the problem indirectly. They look for alternative ways to scale up the exercise without disaggregation. The Command Forces (CFOR) approach is to represent military decision makers, or command entities, as simulation entities (Seidel, Salisbury, Booker, and Dahmann, 1995). Command entities are aggregate entities that control a group of individual entities via exchange of orders and reports through CCSIL messages. Computer resources are freed up by the increased level of autonomous decision making possible with command entities, thereby reducing the number of personnel required to operate a training exercise.

The second approach, UNIFY, is one in which each simulated object maintains information at all required levels of resolution and provides information on demand at the requested level of resolution (Natrajan and Nguyen-Tuong, 1995). For example, a battalion unit would maintain information on each of its platoons and each platoon would maintain information on each of its individual tanks. The problem of aggregation/disaggregation is transformed into a task of serializing and handling each request to a simulated object fully before processing another request. An object decides at what resolution level it wants to preceive another object and a preceived object can consistently present views of itself at different levels of resolution.

2.2.6 Combat Results Correlation Error

A research issue for aggregate and entity linkage systems is the Combat Results Correlation Error (CRCE). The difference in combat results from aggregate and entity level simulations is CRCE and is due to differences in scenario details, terrain models, performance models, behavioral models, force resolution, modeling methodology, level of force aggregation, model time step size, physical, sensory, and behavioral responsivenesss, selection of critical events, and parameter resolution and accuracy and many other factors. The most detailed resolution is provided by the entity level simulation, in which actions of platform and fire-team level battlefield entities are individuallly simulated. By comparison, aggregate level simulations do not simulate but predict combat results from average behaviors of collective forces, ignoring individual entity actions. Such simulations require less data and computing resources than entitiy level simulations. They are familiar to military planners and produce acceptable results for large-scale engagements.

2.2.6.1 Definition: Force Resolution Correlation Error

The FAS report focused on five major, separable sources of CRCE. The approach to constructing detailed definitions of these five CRCE components is the same for each error class. For each component, we identify the relevant submodels in the aggregate and entity simulations, and describe how to modify them to a reference form for which the contribution to CRCE should be very small. The actual CRCE component is then defined as the increase in CRCE that occurs when the component reference submodels are replaced by the component submodels of a practical aggregate (A) plus entity (E) simulation linkage. As the consistency of the linked models is improved by adjusting parameters of the componenet submodels, this shows how close the system comes to the "best possible match."

Entity level simulations account for the motion and engagements of individual entities, while aggregate simulations model only collective motions and encounters of larger units. If a unit too small to be modeled in the aggregate simulation takes actions that makes a large change in the combat results, a large component of CRCE is created that is classified as force resolution correlation error (FRCE). FRCE differs from

Performance Correlation Error (PCE) component of CRCE in that FRCE is an error related to level of disaggregation used in the simulation, while PCE is related to the type of performance model employed, independent of the aggregation level.

Two phenomena related to differences of aggregate and entity force resolution lead to CRCE. The first results from stochastic processes used in entity modeling of force interaction models. The second arises from differences in model resolution. We define only the second phenomenon to constitute FRCE.

Selection of a particular level of detail is a matter of policy related to the purpose of the simulation and available resources. Note that, just as aggregate models can be used at different resolutions (e.g., Corps Eagle vs. Division Eagle), so can entity models. Existing entity systems (e.g. ModSAF) occasionally model aggregations of forces as single battlefield entities (e.g. infantry fire squads). In CLCGF we have modified ModSAF to provide support for general entity models of larger aggregate units (e.g., tank platoons, companies). Let (R_A,R_E) be the resolutions employed normally in the models (A,E). Let $r_E(R_A)$ represent the adaptation of E to function at force resolutions R_A (e.g., battalion). Comparison of (A,E) outcomes in the case of $(R_A, r_E(R_A))$ provides the minimum controlled contribution of FRCE to CRCE. FRCE is the increase in CRCE when the resolution of E is changed from $r_E(R_A)$ to R_E . Values of R in the range from $r_E(R_A)$ to R_E (e.g., companies, platoons), allow dependence of FRCE on resolution to be studied.

2.2.6.2 CRCE Summary

The ideal system would be able to simulate combat in either the aggregate world or in the virtual world and arrive at the same outcome. It is not possible to completely eliminate differences between the two models because aggregation eliminates information. If the information lost during aggregation is not necessary for the subsequent use, the two models could be consistent and maintain face validity.

CRCE undermines user confidence in combat results produced by simulations that link aggregate and entity models. If the combat results of a simulation vary as the simulation shifts from the aggregate to the entity domain, and if this shifting is a simulation artifact, the combat results will be unstable under different assumptions and unreliable.

Validity of aggregation will fail if consistency between combat occurring within a virtual representation and within an aggregate representation is defined as: the outcome of a single scenario should produce similar results. If instead we

- calibrate the aggregate models with appropriate statistical averages over scenarios sampled from large domains of entity variables,
- valid aggregation requires information not maintained by the entity model, e.g. mission objects, strategies,
- systematic bias can be introduced if disaggregation always managed in the same way, e.g. entities disaggregate into same formation,
- recognize that CRCE does exist, and
- apply the aggregate model to combat situations that the model is statistically based on,

then aggregation approximations will be reasonably accurate.

2.3 Summary

A long term goal of systems that link aggregate and entity models is to provide a seamless design so that objects in the simulation can change resolution without disruptions and with some confidence that the results are valid (Davis and Hillestad, 1993). The different levels of detail of the two resolutions make a fair fight difficult (Smith, 1995). Hence, current implementations handle combat by disaggregation of the lower resolution object. Automatic disaggregation/reaggregation is based on an object's location on the terrain or on the nearness of two objects of different resolution. Information about an aggregate unit, such as, location of entities, strength or damage of entities, and fuel level of entities, can be distributed to its individual entities in multiple ways and must flow between the aggregate and entity objects. Current areas of research are managing spreading disaggregation, controlling and minimizing CRCE, and extending the DIS protocol with an aggregate PDU that meets the needs of aggregate+virtual simulations

3. Experimental Design and Analysis

3.1 General Methodology

FRCE is one component of CRCE. Other components are: scenario, terrain representation, performance, and behavioral. The strategy to measure FRCE effect on CRCE is to conduct an experiment where factors that do not comprise FRCE are held constant. Ideally two simulations models will differ only in factors that model SAF behavior as an aggregate unit and in factors that model entity level behavior. For example, movement on terrain, line of sight calculations, combat damage are determined individually by each entity in the virtual simulation run, while in the constructive simulation these computations are determined by algorithms that model aggregate units made up of multiple entities.

Using FRCE component of CRCE to compare the ModSAF (V) system with the constructive Eagle system is not possible. Aggregate units in the Eagle system can only travel on roads, while in ModSAF (V) the units travel cross country. This affects the terrain component of CRCE. The behavioral component of CRCE is affected by the fact that Eagle and ModSAF (V) have different rules of engagement. Eagle units can even automatically withdraw while ModSAF (V) units cannot.

In order to compare two simulation alternatives using the FRCE measure it was decided to develop a ModSAF (C) system. Previous work had been done by CLCGF system. They had developed an aggregate unit that exists in the simulation world, that is it ticks, and it moves. What was missing from CLCGF was the aggregate unit's ability to engage in combat. This requires that an aggregate unit has, for example, recognition capability of opposing forces, weapons capability, and combat attrition capability. With this added capability it is possible to study FRCE by exercising ModSAF (V) and ModSAF (C).

The basic scenario chosen was one where a force holds a defensive position while an opposing force attacks that defensive position. Company size units were chosen because Eagle can not model smaller SAF units. The force ratio is set to 3 to 1. With this basic scenario, combat results for a battle among aggregate units can be computed from well known Lanchester equations. The analysis procedure using Lanchester equations is discussed in Section 3.2. The corresponding scenario for entities utilizes damage tables with a random component to determine the outcome of one to one combat.

Lanchester equations model the paired scenarios: friendly forces defending with opposing forces attacking and the reverse scenario, opposing forces defending with friendly forces attacking. Therefore, the basic scenario was implemented as two scenarios. A further expansion of the basic scenario was made in order to study the SCE component of CRCE. The scenario modification was very slight in order to perturb the simulation situation in a controlled way and to just affect the SCE component of CRCE. It was decided that changing the number of tanks in a company was a way to accomplish this. Four force levels were chosen for each of the two basic scenarios. This resulted in a simulation experiment of eight scenarios for the two ModSAF systems.

The original scenarios implemented were used to test the ModSAF (C) code additions and the Lanchester parameter calibration process. They consisted of the basic scenario on flat terrain. The final scenarios implemented for ModSAF (V), ModSAF(C), and Eagle experiments were constructed with SME guidance. Overlays were placed on terrain tactically appropriate for a defensive position with a frontal attack and rich enough to provide plausible battlefield behaviors. The ModSAF (V) and (C) used the same overlay. The Eagle scenario could not be placed on the same terrain since the attacking force needed to travel on a road. See Appendix D for screen prints of the ModSAF (V) and (C) scenarios.

Experimentation with Force Resolution component of CRCE measurement is conducted as follows:

- Construct a sample scenario.
- Divide the scenario into illustrative vignettes.
- Establish isomorphic aggregate and entity level vignettes.
- Select aggregate attributes to include in core CRCE
- Add aggregate objects to ModSAF code.
- Verify equivalence of force actions.
- Perform sensititvity analysis.
- Evaluate effects on secondary measure of CRCE.

3.2 Calibration Procedure for Aggregate Level Simulation

The basic concept of calibration is to adjust the input parameters to the aggregate-level ModSAF so as to minimize the overall CRCE between the results of virtual- and aggregate-level ModSAF. As mentioned earlier, the CRCE is taken to be the sum of the errors in predicting Enemy and Friendly survivors, averaged over all force-structures, scenarios, and repetitions. This error is minimized when the Lanchester parameters summarizing the overall virtual-level results match those summarizing the overall aggregate-level results.

The calibration procedure is designed in such a way as to minimize CRCE as well as ensure that only FRCE is being measured (not SCE, TCE, BCE, or PCE). The SCE and TCE factors are held constant by using identical terrain and schemes of maneuver. Literally the same datafiles are used by both aggregate- and virtual-level ModSAF. The BCE and PCE sources of error would both manifest themselves as errors in the Lanchester coefficients of the aggregate model that would cause high CRCE - and the minimization process eliminates that source by adjusting the coefficients. Thus, only FRCE remains. Figure 1 describes this calibration process.

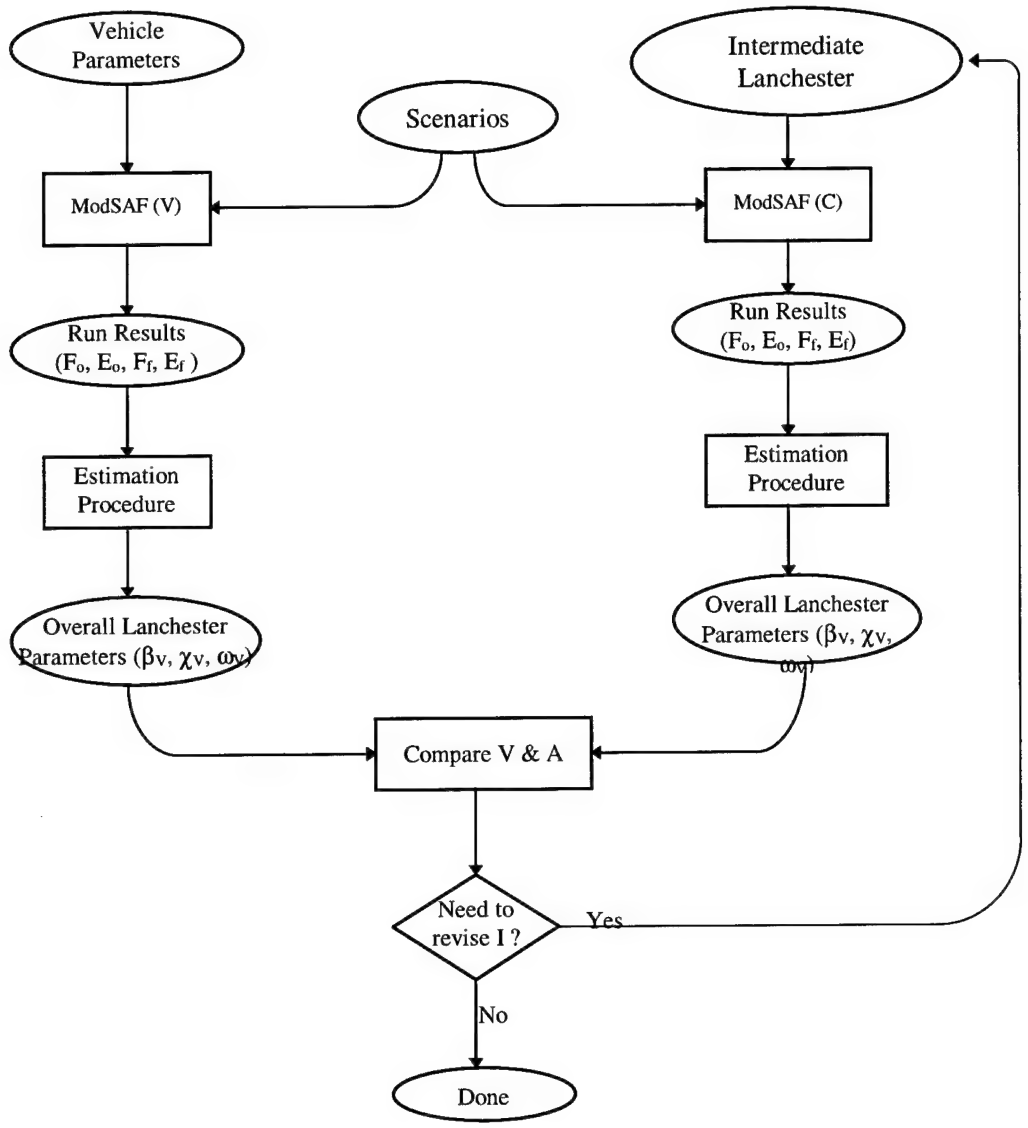


Figure 1: Calibration Process

3.2.1 Lanchester Parameters β and χ for the Two Scenario Case

Lanchester equations are a set of differential equations for modeling warfare. These equations describe the rate of attrition for forces at a given time. The form of the model varies with the scenario under study. For the case where one force is in the defense and the other force is attacking, the Lanchester equations are:

when the friendly forces are in the defense and the enemy forces are attacking:

1) $$\frac{dF}{dt} = \frac{-\alpha_E E_t}{\beta}$$

2) $$\frac{dE}{dt} = -\alpha_F F_t$$

when the enemy forces are executing a defensive action and the friendly force is attacking:

3) $$\frac{dE}{dt} = \frac{-\alpha_F F_t}{\beta}$$

4) $$\frac{dF}{dt} = -\alpha_E E_t$$

where

β	=	defender's advantage
α_E	=	kills per second for the scenario where enemy forces attack friendly forces
α_F	=	kills per second for the scenario where friendly forces attack enemy forces
E_t	=	number of combat capable enemy forces at time t
F_t	=	number of combat capable friendly forces at time t.

The above equation pairs can be solved for *dt* and integrated over the engagement time interval, *t = (0,f),* with *t = 0* is the engagement start time and *t = f* is the engagement finish time. Equations 5 through 9. show the derivation for the case of friendly forces in the defense:

5) $$dt = \frac{\beta dF}{-\alpha_E E_t} \quad and \quad dt = \frac{dE}{-\alpha_F F_t}$$

Substituting *dt* and integrating, the following equations result:

6) $$\alpha_F F_t dF = \tfrac{\alpha_E}{\beta} E_t dE\,,$$

7) $$\alpha_F \int F_t dF = \tfrac{\alpha_E}{\beta} \int E_t dE\,.$$

Integrating both sides over the interval *t=(0,f)* leads to a set of equations called the Lanchester Square equations:

for case with friendly forces defending

8) $$\alpha_F F_t^2 = \tfrac{\alpha_E}{\beta} E_t^2\,,$$

for case with friendly forces defending

9) $$\tfrac{\alpha_F}{\beta} F_t^2 = \alpha_E E_t^2\,.$$

If we integrate over the engagement time interval, t = (0,f), we can describe combat attrition using the Lanchester square equations as follows:

for the case where the friendly forces are defending:

10) $$\alpha_F(F_0^2 - F_f^2) = \tfrac{\alpha_E}{\beta}(E_0^2 - E_f^2)$$

for the case where the enemy forces are defending:

11) $$\tfrac{\alpha_F}{\beta}(F_0^2 - F_f^2) = \alpha_E(E_0^2 - E_f^2).$$

Now, define variables G and K, with subscript A representing the attacking force and D the defending force, to be the normalized difference of the squared force levels by dividing the difference by the total losses of both forces:

for the friendly forces in the defense scenario

12) $$G_A = \frac{E_0^2 - E_f^2}{(E_0^2 - E_f^2) + (F_0^2 - F_f^2)}$$

13) $$K_D = \frac{F_0^2 - F_f^2}{(E_0^2 - E_f^2) + (F_0^2 - F_f^2)}$$

for the enemy forces are in the defense scenario

14) $$G_D = \frac{E_0^2 - E_f^2}{(E_0^2 - E_f^2) + (F_0^2 - F_f^2)}$$

15) $$K_A = \frac{F_0^2 - F_f^2}{(E_0^2 - E_f^2) + (F_0^2 - F_f^2)}$$

where $G_A + K_D = 1$.

Using this form of the equations we can define two Lanchester parameters, β and χ, in terms of force levels at the start and end of an engagement. β is defined above as the defender's advantage and χ is the ratio of the each force's attrition rate parameter α. That is,

16) $$\chi = {}^{\alpha_F}\!/_{\alpha_E}.$$

Rewriting in terms of these new variables we have:

finally for the case of friendly forces in the defense:

17) $$G_A = \beta\chi K_D$$

and for the case of enemy forces in the defense:

18) $$G_D = {}^{\chi K_A}\!/_{\beta}.$$

Using equations 17 and 18 we solve for χ and β:

$$19)\quad \chi = {\beta G_D}/{K_A}$$

Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.

$$20)\quad \beta = \sqrt{\left({G_A K_A}/{G_D K_D} \right)}$$

Equations 19 and 20 hold for the case where there are two scenarios: friendly forces in the defensive with enemy forces attacking and friendly forces attacking with enemy forces in the defense.

3.2.2 Overall Lanchester Parameters β and χ for the Many Scenario Case

Generalizing, let there be *n* scenarios with the friendly forces in the defensive and the enemy forces attacking and *m* scenarios with the friendly forces attacking and the enemy forces in the defense. Then we have n+m Lanchester Square equations as found in Equations 8 and 9. Rewriting those equations in terms of β and χ we have equation of the form:

for the case with friendly forces defending

$$21)\quad \chi\beta F_{t_i}^2 = E_{t_i}^2 ,$$

for the case with friendly forces defending

$$22)\quad \tfrac{\chi}{\beta} F_{t_j}^2 = E_{t_j}^2$$

where $t_k=(s,f)$ is the engagement duration interval for each equation. These equations are now in the form to apply linear regression. Linear regression estimates for the coefficents are:

For the case where the friendly forces are defending:

$$23)\quad \mathrm{X}_D = \chi\beta = \frac{\sum_{i=1}^{n}\left(F_{t_i}^2 - mean\left(F_{t_i}^2\right)\right)\left(E_{t_i}^2 - mean\left(E_{t_i}^2\right)\right)}{\sum_{i=1}^{n}\left(F_{t_i}^2 - mean\left(F_{t_i}^2\right)\right)^2}$$

For the case where the enemy forces are defending:

$$24)\quad \mathrm{X}_A = {\chi}/{\beta} = \frac{\sum_{j=1}^{m}\left(F_{t_j}^2 - mean\left(F_{t_j}^2\right)\right)\left(E_{t_j}^2 - mean\left(E_{t_j}^2\right)\right)}{\sum_{j=1}^{m}\left(F_{t_j}^2 - mean\left(F_{t_j}^2\right)\right)^2}$$

We can solve for β and χ by:

$$25)\quad \mathrm{X}_D \mathrm{X}_A = (\chi\beta)\left({\chi}/{\beta}\right) = \chi^2 \text{ and then}$$

26) $$\beta = \sqrt{X_D / X_A}$$

27) $$\chi = \sqrt{X_D X_A}\,.$$

This time β is the *overall* defender's advantage and χ is the *overall* ratio of the each force's attrition rate parameter α.

3.2.3 Overall Lanchester Parameter ω

χ measures the relative strength of friendly to enemy forces. Parameter ω is related to the duration of the battle and its intensity over time. This parameter is related to χ and is defined as:

28) $$\omega = \alpha_F \alpha_E$$

Using equations 16 and 28 the *overall* α_F and α_E can be solved for:

29) $$\alpha_F = \sqrt{\chi\omega}$$

30) $$\alpha_E = \sqrt{\omega / \chi}$$

Equations 19 and 20 define β and χ in terms of friendly and enemy forces levels. Equations29 and 30 can now be used to define ω in terms of the forces levels. Starting with the Lanchester equations1, 2, 3, and 4, but this time as difference equations, we have:

when the friendly forces are in the defense and the enemy forces are attacking:

31) $$\Delta F_i = \frac{-\alpha_E E_i}{\beta} \Delta t$$

32) $$\Delta E_i = -\alpha_F F_i \Delta t$$

when the enemy forces are executing a defensive action and the friendly force is attacking:

33) $$\Delta E_j = \frac{-\alpha_F F_j}{\beta} \Delta t$$

34) $$\Delta F_j = -\alpha_E E_j \Delta t$$

Substituting the α's with χ and ω, we estimated the force levels:

when the friendly forces are in the defense and the enemy forces are attacking:

35) $$\Delta \hat{F}_i = -\sqrt{\omega}\left(\frac{E_i \Delta t_i}{\beta\sqrt{\chi}}\right)$$

36) $$\Delta \hat{E}_i = -\sqrt{\omega}\left(F_i \Delta t_i \sqrt{\chi}\right)$$

when the enemy forces are executing a defensive action and the friendly force is attacking:

$$37) \quad \Delta\hat{E}_j = -\sqrt{\omega}\left(\frac{F_j \Delta t_j \sqrt{\chi}}{\beta}\right)$$

$$38) \quad \Delta\hat{F}_j = -\sqrt{\omega}\left(\frac{E_j \Delta t_j}{\sqrt{\chi}}\right)$$

Equations 35, 36, 37, and 38 can be rewritten into the following simplified form:

$$39) \quad \theta_k = \sqrt{\omega}\lambda_k$$

by defining the following symbols:

λ_k equals the factors in the large parenthesis for the kth equation ,
θ equals the change in the force level for the kth equation,
and $1 \leq k \leq 2(n + m)$ for *n* scenarios with friendly forces defending and *m* scenarios with enemy forces defending.

The 2(n + m) equations can be solved for $\sqrt{\omega}$ by using the least squares method and results in:

$$40) \quad \sqrt{\omega} = \frac{\sum_{k=1}^{2(n+m)} \theta_k \lambda_k}{\sum_{k=1}^{2(n+m)} \lambda_k^2}$$

3.2.4 Iterative Process for Estimating and Adjusting the Lanchester Parameters

Using the process defined above, the parameter triple (β,χ,ω) can be calculated given the force levels at the start and end of each combat. In particular, equations 19, 20, and 40 are used to compute the parameters that summarize the overall results from all combats. These three parameters describe the overall set of scenarios, and they are applicable to virtual ModSAF, constructive ModSAF, and Eagle: they can be used to describe the overall behavior, whether the lowest level phenomena are described by Lanchester equations, entity-to-entity engagements, or some other process.

In the case of ModSAF(V), each scenario is executed repeatedly. Because of the randomeness included in ModSAFEs combat tables, the damage caused by each firing of a weapon is not precisely duplicated every run. As the entity-level parameters of ModSAF (V) are not changed, the parameters (β_V,χ_V,ω_V) describing its behavior over all the runs need only be computed once.

ModSAF(C) does use Lanchester coefficients (β_I,χ_I,ω_I) to describe the company-to-company level combat. However (just like ModSAF(V)) the phenomena of multiple simultaneous engagements, maneuver, etc. are present, so the overall parameters (β_A,χ_A,ω_A) describing the overall behavior will be different from the company-to-company coefficients. Thus, (β_I,χ_I,ω_I) will need not only a starting value, but they will need to be adjusted until the resulting (β_A,χ_A,ω_A) are close to (β_V,χ_V,ω_V).

This adjustment process is as follows:

Let γ, η, and φ be the error in calculating β, χ, and ω from observations, $i = 1, .., n,$ output by a simulation using $(\beta_I,\chi_I,\omega_I)$ as an inital value or guess for the Lanchester parameters. Then,

$$41)\quad \beta_A = \beta_I \gamma$$

$$42)\quad \chi_A = \chi_I \eta$$

$$43)\quad \omega_A = \omega_I \phi$$

Let the next guess for β, χ, and ω be derived from $(\beta_V,\chi_V,\omega_V)$.:

$$44)\quad \beta_{I+1} = \beta_V / \gamma$$

$$45)\quad \chi_{I+1} = \chi_V / \eta$$

$$46)\quad \omega_{I+1} = \omega_V / \phi$$

Then elimating γ, η, and φ from the above equations we have:

$$47)\quad \beta_{I+1} = (\beta_V * \beta_I) / \beta_A$$

$$48)\quad \chi_{I+1} = (\chi_V * \chi_I) / \chi_A$$

$$49)\quad \omega_{I+1} = (\omega_V * \omega_I) / \omega_A$$

The adjustment procedure stops when the following criterion is met for a given ε:

$$50)\quad \left(\frac{(\beta_{I+1} - \beta_V)}{\beta_V}\right)^2 + \left(\frac{(\chi_{I+1} - \chi_V)}{\chi_V}\right)^2 + \left(\frac{(\omega_{I+1} - \omega_V)}{\omega_V}\right)^2 \leq \varepsilon$$

3.3 Scenario Description

The Scenario was done on the Hunter-Ligget database. This database was chosen since it was known to be available to both ModSAF and to Eagle, which was required for comparison purposes.

The Scenario(s) were enemy forces on friendly forces and friendly forces on enemy forces, always one company in defense and three companies attacking. The attacking companies would approach in column on three separate parallel routes, and would deploy into line formation when the assault was expected. Effective engagements usually started at a distance under 3500 meters. See Appendex D for three views of the ModSAF exercise: start of ModSAF (V), start of ModSAF (C), and combat during a ModSAF (V) execution.

The defending units were placed in "defensive" positions on the front of a small hill. The assaulting units would come up through the defending units to a final position on the back side of the hill. On occasion there would be defending units that would survive the engagement on the front of the hill. There was also the Command elements of the defending company back from the front of the defensive position, and these elements would engage the remaining assaulting elements that came into it's limited field of fire (this was in some trees) as they came into their final positions, thus continuing the engagement a bit longer.

The ModSAF Constructive scenario used the same graphics that were created for the virtual scenarios, and so operated over the exact same terrain those scenarios used. To run the enemy on friendly and friendly on enemy scenarios with variations in the vehicle count for each side at the Virtual level required each pairing to have it's own unique scenario. The ModSAF Constructive implementation determines how many vehicles compose the Units by looking at the data file, so there were only two scenarios required for the Constructive tests, one for red assaulting, and one for blue assaulting, and the vehicle numbers were varied by altering the data file for each run.

3.4 Aggregate Attributes Included in FRCE

Two types of data are collected from ModSAF (V), (C), and Eagle experiments. The first type is force level. The force level at the start of the simulation and at the end of combat are recorded. Then attrition is calculated by subtracting the final level from the initial level. Attrition is a value that is clearly identifiable and easy to interpret. It is also is measurable and not artificial. Force level values are easily collected from both ModSAF and Eagle simulations.

Attrition values are used to calibrate the ModSAF (C) simulation and to measure FRCE and SCE. A complete presentation of the procedure for calibration of ModSAF (C) can be found in Section 3.2. The basic procedure for calibrating the ModSAF (C) is:

1. Run the ModSAF (V) experiments and calculation the Lanchester parameters, (β,χ,ω).
2. Using the Lanchester parameters, (β,χ,ω), from step 1 as input to ModSAF (C).
3. Run the ModSAF (C) experiments and calculation the Lanchester parameters, (β,χ,ω).
4. Compare the Lanchester parameters from step 1 with those from step 3.
5. If the parameters from step 3 are not close enough, then adjust the step 3 parameters, use the adjusted values as input to ModSAF (C), and return to step 3.
6. When the values are close enough use the attrition values from step 1 and the final execution of step 3 to compute FRCE and SCE.

The second type of data that is collected from the simulation runs is engagement time. Engagement time is used in the calculation for the Lanchester parameter, ω. This value is captured by recording the simulation time when combat begins and ends. Commencement of combat is defined as the time the first weapon is fired. The end of combat is defined as either when one side's forces are all damaged in such a way as to inhibit combat or when the attacking unit has passed through the defender's position and no more hits are possible. For example, a tank that has a mobility kill still has combat capability.

3.5 Modification to ModSAF Code

3.5.1 Modifications to CLCGF Code

One system has been developed for LCVS. This system's code is an extension of CLCGF 2.0D which uses the ModSAF 2.0 baseline. Both ModSAF (V) and (C) can be executed using the same executable but with different input scenarios defining the unit type. ModSAF (V) was already available in CLCGF. CLCGF had an aggregate unit capability, but with minimal functionality. The aggregate unit could not participate in combat. The following modifications were performed on the CLCGF ModSAF code.

There already existed a basic Aggregate simulation model in the CLCGF ModSAF used as the baseline. However, this model had no ability to interact and inflict, or take, damage. Lanchester equation implementations were added to *libsrc/libaggunit* to accomplish this requirement. These equations were initialized out of a new datafile that was added to the library. This allows modification of the parameters to the aggregates without having to modify the code and recompile.

The ability to do a data-dump of both the Aggregate and Entity simulation runs was added. All factors that went into the aggregate equations were added to the Aggregate dump to aid in the record keeping of where the particular results came from. The Entity data-dump contained simply the results of the test (Enemy and friendly strength at start and end), since there were not going to be any modifications of the factors determining the outcome of the Entity level scenarios. This datafile dump capability can be controlled from the ModSAF parser.

A system was implemented that would monitor the scenario, and would detect when the aggregate run was done (when all of one side was attrited completely). This would trigger a data dump, and shut the ModSAF process down. A similar system was implemented for the entity level runs.

A timing tool was implemented in the simulation that would alter the simulation clock to maximize the rate at which the simulation ran. This tool monitored the system load, and either sped up or slowed down the Simulation clock as load decreased or increased. This allowed the average entity Scenario run to take less than half the time it otherwise would have, and allowed the Aggregate runs to take a fraction of the time they otherwise would have. It also guaranteed that the system would not overload, throwing question on the results.

3.5.2 Lanchester Models in ModSAF (C)

The ModSAF (C) code was implemented with the Lanchester model as found in Equations 1-4. The Eagle system uses Vector Lanchester Equations from Vector Research Company.

3.5.3 Software Tool for Mathematical Calculations

Do-Math is the name for the software tool developed to do the mathematical calculations for determining the new estimation for Defender's Advantage, Ratio of Lethality, and from these the attrition factors for the friendly and enemy units. The mathematics involved are described in Section 3.2.

Do-Math operates on data files that are made up of the data produced by runs of the Modified ModSAF. Particular output data is edited into a particular format in a single data file that will act as the input data file fed to Do-Math. There are a few scripts that were created to help this process.

Which data file is read in is specified by the user on the Do-Math command line. Information to be operated on is internally grouped by number of red forces and number of friendly forces at the start of the scenario. Each instance with a particular number of friendly and enemy forces is treated as an instance of that particular set. Each instance in the data file is looked through in a first pass to determine how many instances of each set exist. This number is then used in the second pass to load the data array with the same number of instances from each data set so that there will be a balanced number of each set represented in the calculations.

3.5.4 Standalone, non-DIS Version

For the experimentation required under the LCVS DO, there was no need to create a networked version of LCVS's system. The experimentation aspect of this work did not include implementing extensions to the DIS PDU protocol for transmitting aggregate state information. A discussion of the aggregate DIS PDUs necessary for implementing a networked solution is found in Section 2.2.4.

3.6 ModSAF (V) Experiments

Once the scenarios have been extablished the next step is to run the entity level simulation, ModSAF (V), in order to calculate the overall Lanchester parameters β and χ. In Section 3.2.1 on Lanchester parameters the Lanchester Square equations are established.

Let X_A and X_D represent the Lanchester parameter ratios as in equations 23 and 24. The data input to each linear regression equation is the 30 runs at each of four force structures (and one regression equation for each posture). From this data, X_A, X_D, β, and χ are calculated to be the following:

$X_A = 0.8110$

$X_D = 0.1028$

$\beta_V = 2.8088$

$\chi_V = 0.2887$

3.7 ModSAF (C) Experiments

The next step is to calibrate ModSAF (C) as described in Section 3.2.4. This process, repeated here, proceeds as follows:

1. $(\beta_V,\chi_V,\omega_V)$ is used as the starting value for parameter triple. We call the input triple $(\beta_I,\chi_I,\omega_I)$.
2. Execute ModSAF (C) for all the scenarios with the input triple $(\beta_I,\chi_I,\omega_I)$ and record the force levels.
3. Use equations 19, 20, and 40 to estimate a new triple, $(\beta_A,\chi_A,\omega_A)$, from the ModSAF (C) force level data.
4. Calculate then next input triple, $(\beta_{I+1},\chi_{I+1},\omega_{I+1})$ using equations 47, 48, and 50.
5. Stop when the inequality criterion given in equation number 50 is met for a specified ε.
6. If the criterion is not met, then set the Ith parameter triple value with the I+1th value and repeat with step 2 above.

3.8 Analysis of Variance

Analysis of variance (ANOVA), see Figure 2, reveals that cases of friendly forces defending are much better predicted. The question is how good of an estimator of ModSAF (V) is ModSAF (C). Several measures are considered.

In the following, we use:

2 postures	case are: Friendly defending and Friendly attacking
4 force levels	ratios are: (10,10) (18,10) (14,14) (14,10)
30 runs	for each posture, force level combination.

3.8.1 Measure 1: Analysis of Variance

Given any two simulations, W and Z we can define:

$F_{X_{ijk}}$ = surviving Friendly forces in the ith posture, jth force ratio, kth run, for simulation X.

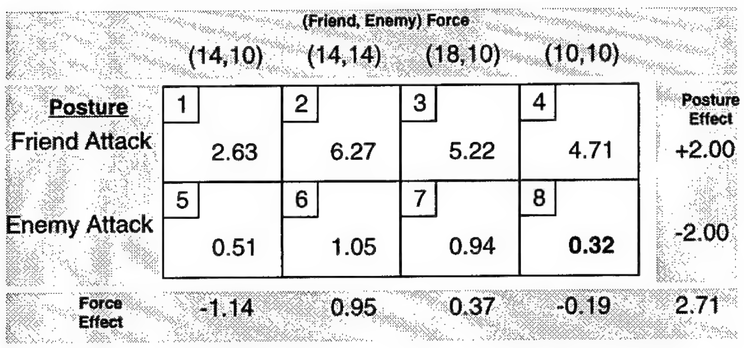
Similiarly, $E_{X_{ijk}}$ is defined for the surviving Enemy forces.

Then the error in outcomes in number of tanks is defined as:

$$D_{ijk} = \sqrt{\frac{\left(F_{W_{ijk}} - F_{Z_{ijk}}\right)^2 + \left(E_{W_{ijk}} - E_{Z_{ijk}}\right)^2}{2}}.$$

For ModSAF (V) versus ModSAF (C) this measure is averaged over the 30 executions:

$$D_{ij} = \sum_k D_{ijk} \Big/ 30 \ .$$



	(Friend, Enemy) Force				
	(14,10)	(14,14)	(18,10)	(10,10)	
Posture					**Posture Effect**
Friend Attack	$\boxed{1}$ 2.63	$\boxed{2}$ 6.27	$\boxed{3}$ 5.22	$\boxed{4}$ 4.71	+2.00
Enemy Attack	$\boxed{5}$ 0.51	$\boxed{6}$ 1.05	$\boxed{7}$ 0.94	$\boxed{8}$ **0.32**	-2.00
Force Effect	-1.14	0.95	0.37	-0.19	2.71

Figure 2: Analysis of Variance Results

Using the linear model:

$$\hat{D}_{ijk} = \eta_j + M + \varepsilon_i + \sigma_{ijk} \, ,$$

we:

main effect, M, equal to 2.706

noise σ equal to 3.913.

This linear model explains 54.0% of the variance in D.

3.8.2 Graphical Analysis of Error Measure D_{ijk}

Further analysis can be done by exploring the data visually. First the means as intervals can be plotted next to each other. In Figure 3 and Figure 4, we can see that the means differ by posture. For enemy forces attacking both simulation models generate root mean squared error term of surviving tanks closer to zero than when friendly forces are attacking.

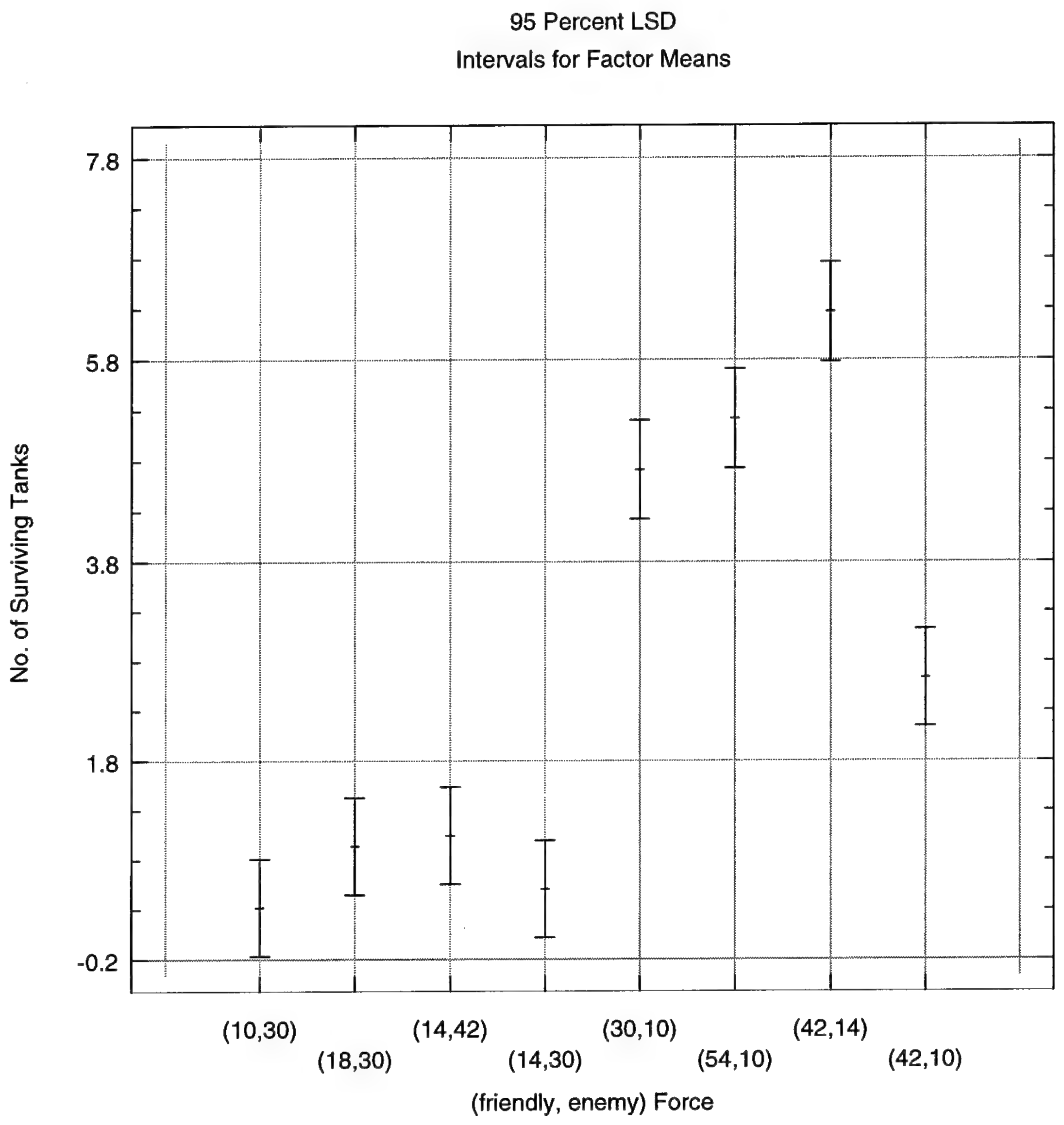


Figure 3: Interval Means for Error Metric, D, Against Force Levels

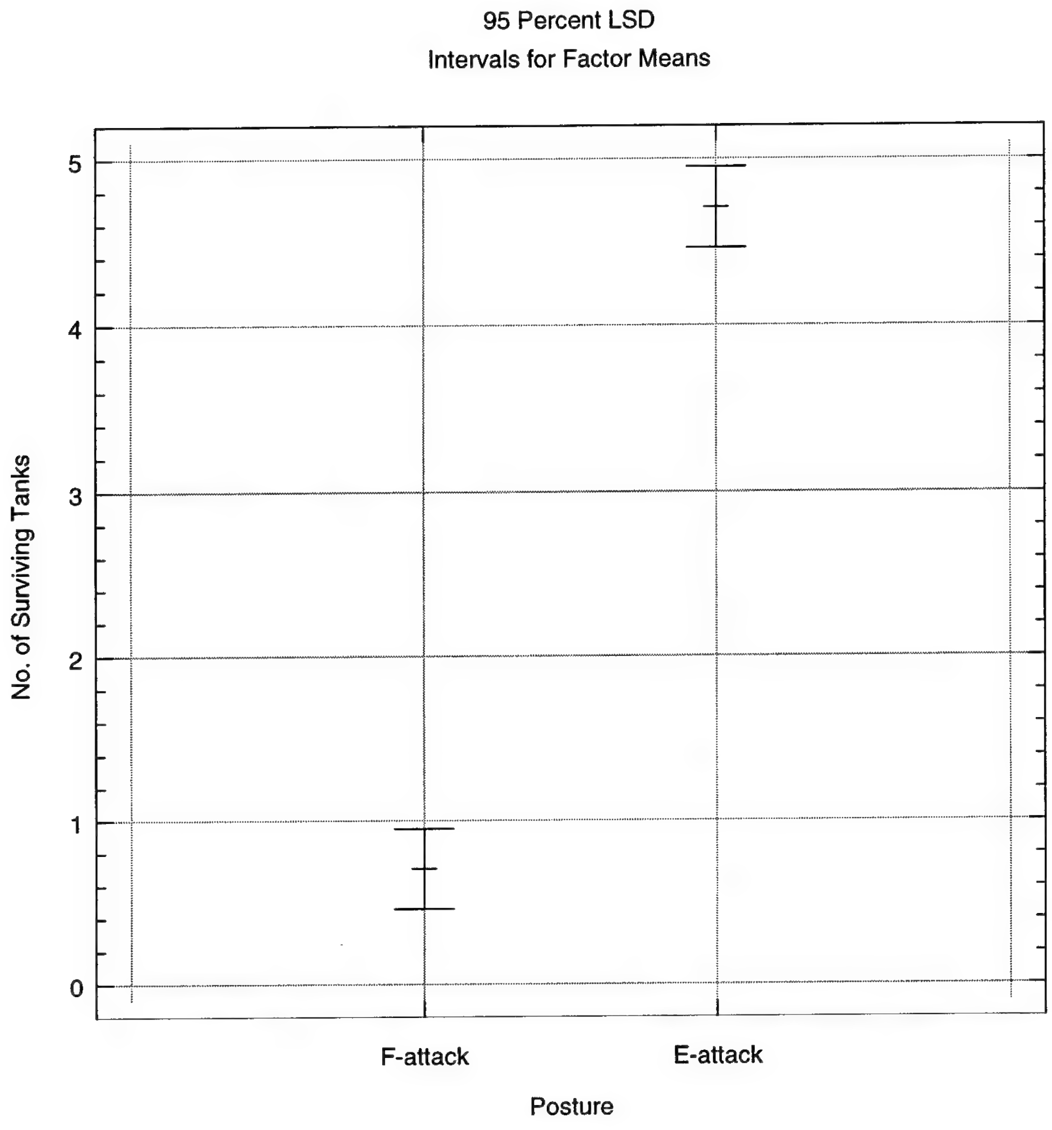


Figure 4: Interval Means for Error Metric, D, Against Posture Levels

Graphical methods allow one to represent data values graphically, without seeing all the details. A box plot is a graphical technique that summaries data into percentiles. The box contains 50% of the values with the data's median denoted as a line through the box. The line above and below the box denote the 10th to 25th and the 75th to 90th percentiles. Data values above 90th or below 10th percentile are plotted as points. Thus the empirical local distribution of the data can be visually studied.

Multiple box plots graphed along side of each other on the same vertical scale allow one to study how a dependent variable changes as a function of an independent variable. A box plot is drawn for each discrete value of the independent variable. Comparing percentiles is an informative way to compare two empirical distributions. In the case of Force Levels, as seen in Figure 5 there are eight discrete values. The four box plots on the left summarize the survival results for the case of the enemy forces attacking. The four plots on the right are for friendly forces attacking. Figure 6 groups the results into the two posture values and we can clearly see that scenarios make a difference. D_{ij}, the error in outcomes in the number of tanks, is closely centered on zero for the case where enemy forces attack and friendly forces are in the defense.

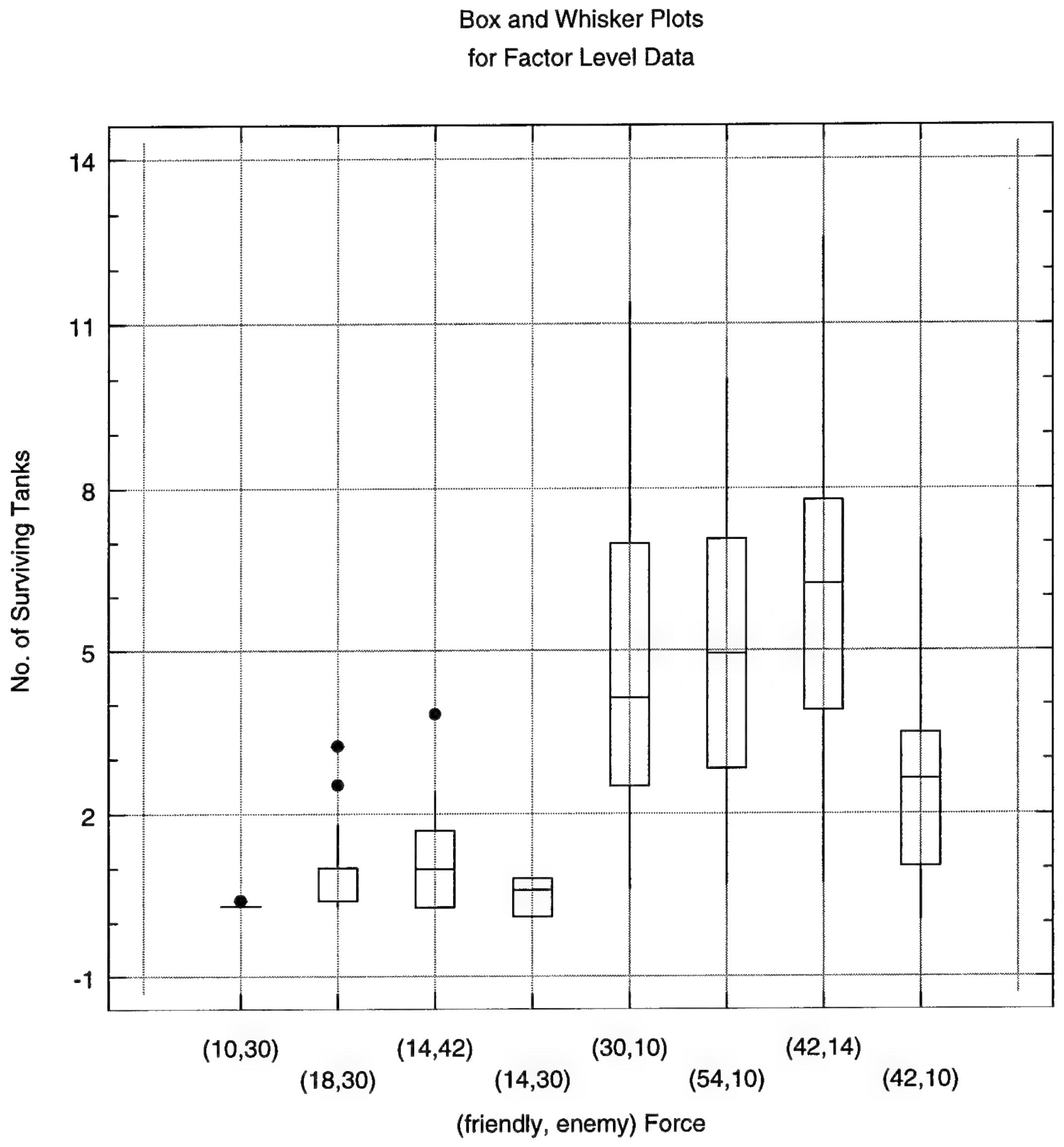


Figure 5: Box Plot Error Metric, D, Against Force Levels

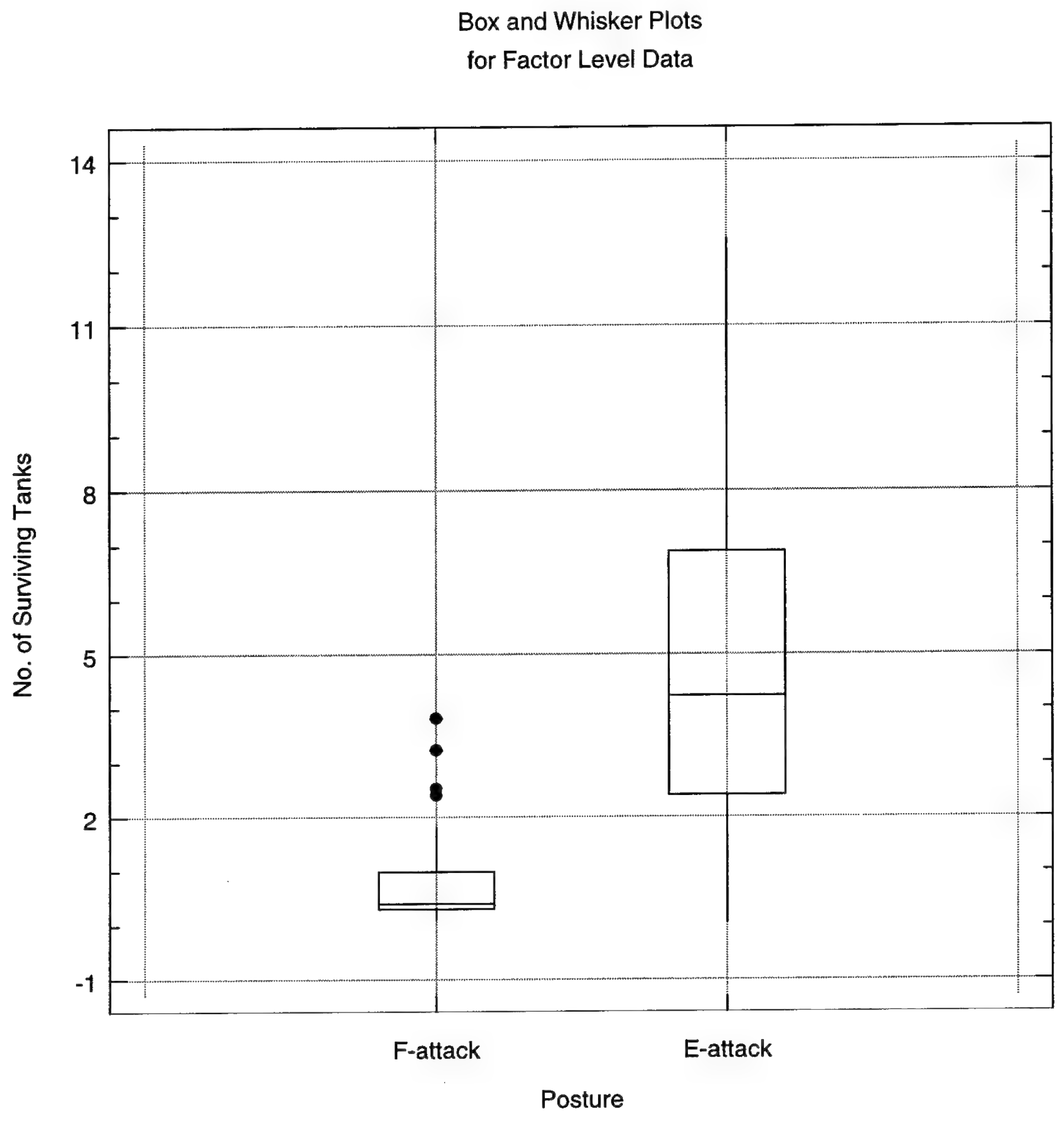


Figure 6: Box Plot for Error Metric, D, Against Posture Levels

The box plots shown in Figure 5 and Figure 6 show scenario effects clearly but not whether aggregate level versus entity level modeling has an effect on the results. One way to examine the effects of the different modeling techniques is to look at paired difference of attrition values. Figure 7 and Figure 8 show:

$$D'_{ijk} = A_{W_{ijk}} - A_{Z_{ijk}}$$

where W and Z are the two simulations, aggregate and entity, and A is the attrition result for the ith posture, jth force ratio, and kth simulation run.

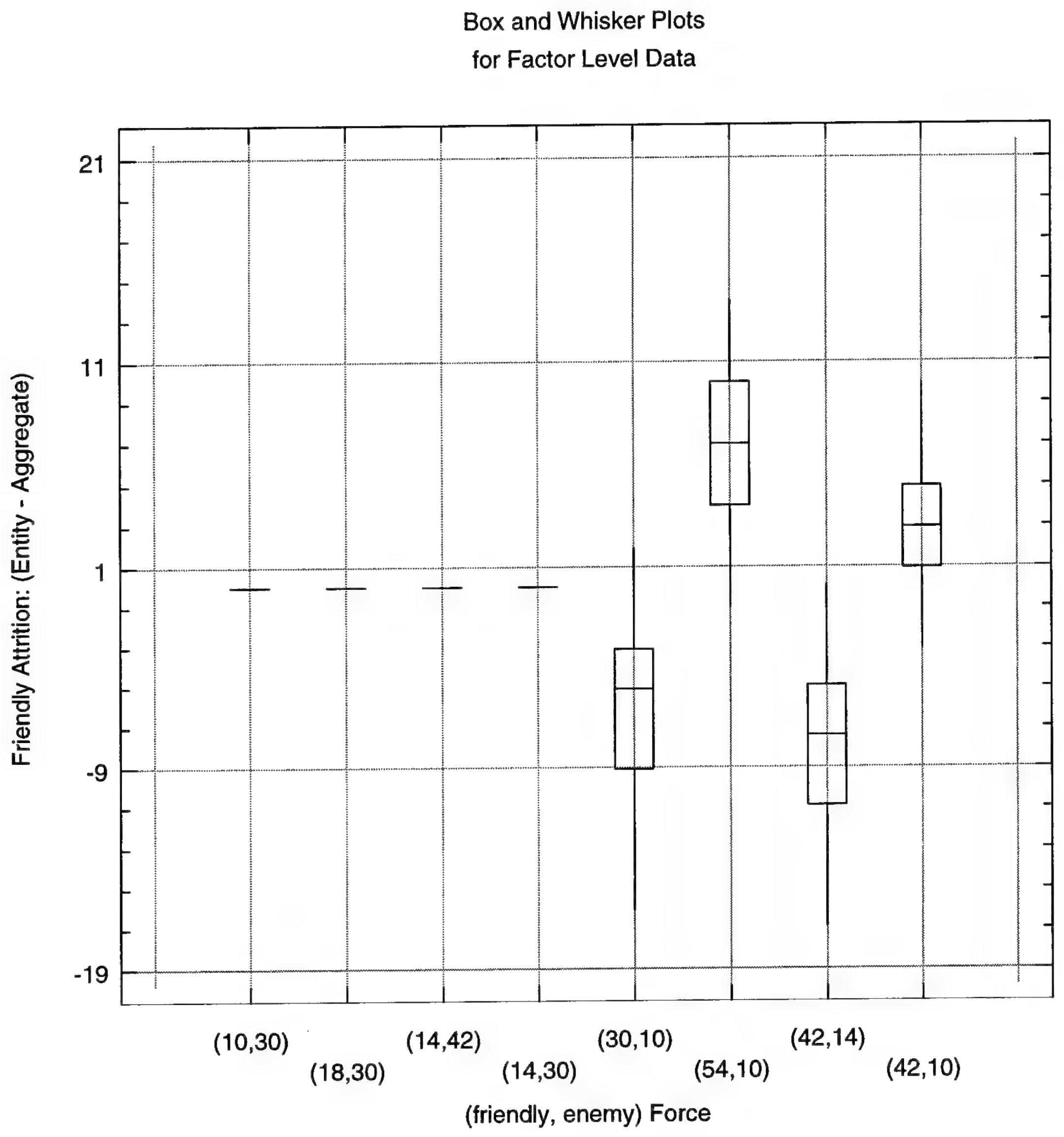


Figure 7: Paired Differences of Aggregate and Entity Values of Attrition of Friendly Forces

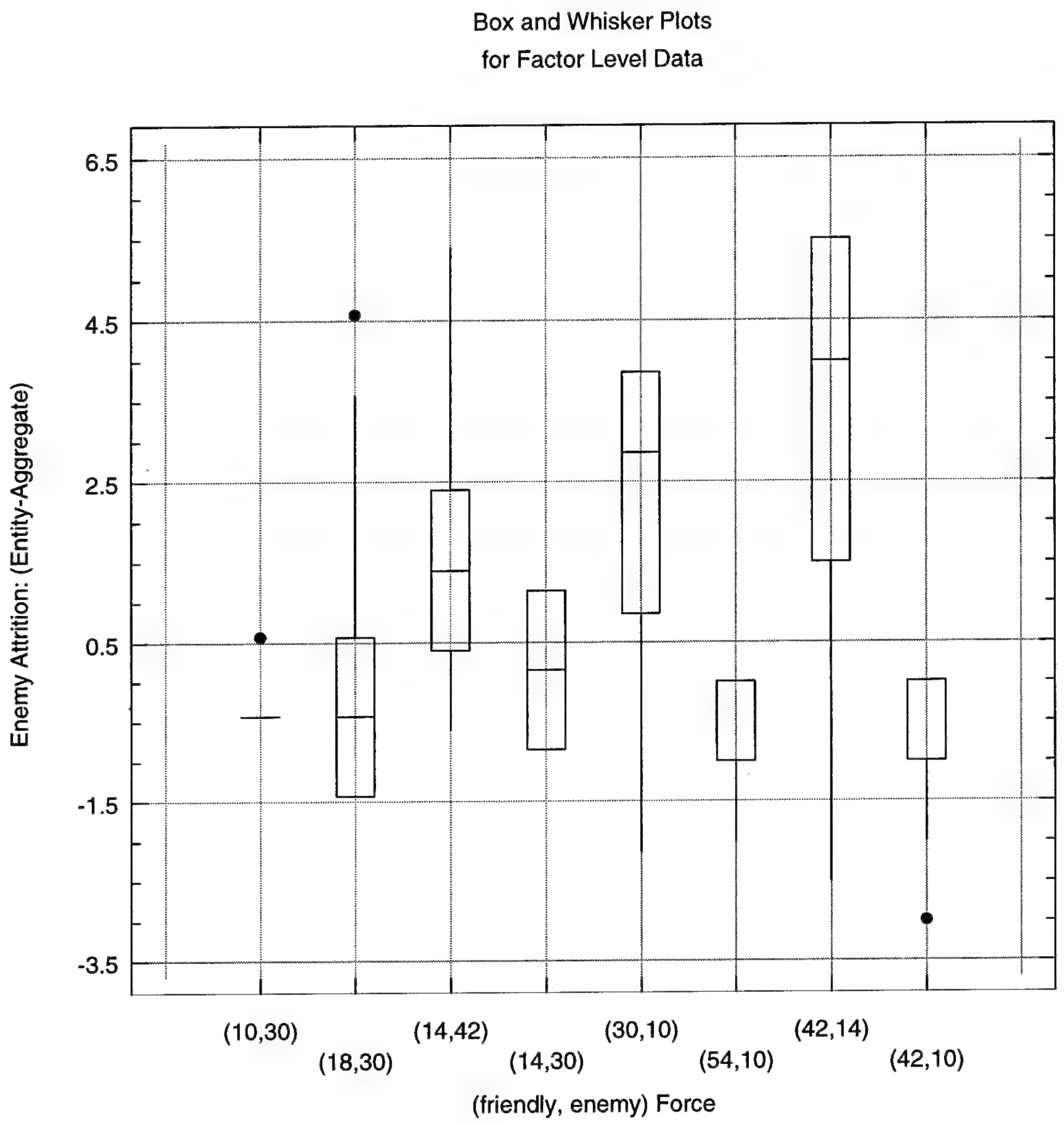


Figure 8: Paired Differences of Aggregate and Entity Values of Attrition of Enemy Forces

Ideally we would like the results to show that the modeling techniques are equivalent and that mean D'_{ijk} is equal to zero. Looking at the box plots we can see that for the case when enemy forces are attacking, friendly attrition is essentially equal for the two simulation models. This is seen by looking at the four plots on the left in Figure 7. Its much harder to make statements about the other plots since they generally overlap along the vertical axis and straddle zero.

These graphical depictions of the simulation data raise new issues that can lead to hypothesis formulation and further studies. Some of these issues are:

- Posture is a factor that clearly has an effect. The effect is most notable for the enemy attrition values.
- The calibration of Lanchester parameters results in prediction of attrition by ModSAF (C) that is "right on the money" for the case of friendly attrition when enemy forces are attacking, but not for enemy attrition and not for the other posture.
- There appears to be a under prediction of enemy combat loses. ModSAF (V) looses more tanks than the Lanchester model predicts will be lost. See Figure 8. This effect is more pronounced for the posture of friendly forces attacking.
- The above effects may partially be due to ModSAF design. For example, ModSAF BLUFOR code/tables may not be equivalent in fidelity to OPFOR code/tables.

3.8.3 Measure 2: Overall Error

Define the overall error between two simulation models, W and Z, to be:

$$D=\sqrt{\tfrac{1}{240}\sum_{ijk}\frac{\left(F_{W_{ijk}}-F_{Z_{ijk}}\right)^2+\left(E_{W_{ijk}}-E_{Z_{ijk}}\right)^2}{2}}.$$

This is the CRCE for the two simulations. This statistic is for paired differences and is a stricker test than if the difference of averages were used.

Again this is the root mean square difference in predicted attrition in force level which is in number of tanks.

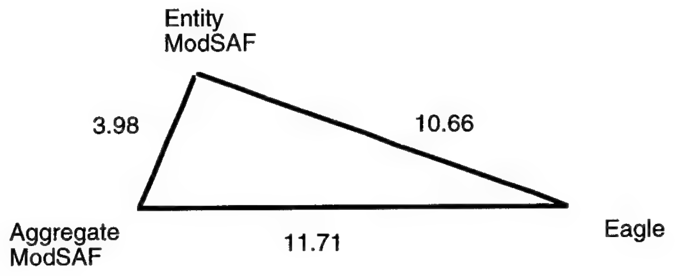


Figure 9: Overall Error

Figure 9 expresses the result graphically. Here FRCE accounts for 37.3% to 34.0% of the total error between ModSAF (V) and Eagle.

3.8.4 Measure 3: Direct Correlation

For the third measure, consider ModSAF (C)to be an estimator of ModSAF (V). We can determine how much of the variance is explained by:

$$1-\frac{unexplained\ \mathrm{variance}}{total\ \mathrm{variance}}=1-\frac{E\left[(V-A)^2\right]}{E\left[(V-\bar{V})^2\right]}$$

where:

V = entity level results = (F_{ev}, E_{ev})

A = aggregate level results = (F_{ea}, E_{ea})

$\overline{V}$ = average of vehicle results.

Using vector subtraction:

$$\begin{aligned} E\left[(V-A)^2\right] &= E\left[\left((F_{ev},E_{ev})-(F_{ea},E_{ea})\right)^2\right] \\ &= E\left[(F_{ev}-F_{ea},E_{ev}-E_{ea})^2\right] \\ &= E\left[F_{ev}^2-2F_{ev}F_{ea}+F_{ea}^2+E_{ev}^2-2E_{ev}E_{ea}+E_{ea}^2\right] \end{aligned}$$

In this sense, ModSAF (C) explains 92.6% of the variance in ModSAF (V). Eagle only explained 46.7% of ModSAF (V).

4. Conclusions

Several conclusions have been reached by this study:

- FRCE was isolated and demonstrated to be roughly 1/3 of the total CRCE between Eagle and ModSAF on this scenario set.
- Aggregate ModSAF was shown to be an accurate predictor of entity ModSAF, after tuning, on this scenario set.

5. Recommendations

Further understanding of the limits and potentials of CRCE control can be achieved by experimentation. The following recommendations and discussion are results of the entire activity of LCVS:

- Apply LCVS methodology:
    - investigate other CRCE components, use other scenarios
    - investigate other Aggregate + Entity linkages.
    - investigate CRCE with a system that controls spreading disaggregation (HySim).
    - search for properties of scenario, terrain, force resolution, performance, behaviors that can identify "safe" multi-resolution interaction areas.
    - search for scenarios set that covers and classifies the simulation space for CRCE control.
- Develop LCVS tools for future use:
    - baseline into ModSAF LCVS code, (Aggregate ModSAF).
    - automate the calibration process.
- Apply LCVS methodology with a system that includes a linkage with Live Simulations.
- Extend ModSAF into a multiple resolution modeling (MRM) system:
    - develop aggregate (move, attrite, sense, consume algorithms).
    - extend interfaces (aggregate interactive protocol, DIS++/HLA FOM).

5.1 Discussion of Development of Aggregate ModSAF

The last item in the above bullet list, extending ModSAF into a MRM system, requires special discussion. The following discusses motives to building such a system and its design.

The simulation community has expressed a desire to significantly expand the possible scale of current simulation scenarios to entity counts of 10,000, 100,000, and beyond. One of the approaches to achieving this goal is to use aggregate simulations to handle areas in the scenario where detail is not required during some or all of the scenario.

The problem with this approach until now has been that using existing aggregate simulations for this purpose has been cumbersome and required significant knowledge of the operation of the aggregate simulation to be used, and different equipment to support it. Also, interacting these simulations with entity simulation (ModSAF) has been cumbersome at best, since methods of operation have been vastly different, protocol support almost non-existent, and existing aggregate simulations were not implemented with interactive/real-time simulation or networked implementation in mind. Also, the separate aggregate implementations have probably not done a decent job of predicting the behaviors and outcomes of a ModSAF based scenario. This would be a desired goal since it is assumed that much of the entity simulations currently being replaced with aggregates would be ModSAF based simulations.

A ModSAF aggregate implementation would solve many of these problems. The behaviors of the ModSAF entity scenarios would map easily/directly to the new aggregate implementation's scenarios, allowing easy transition from one to the other, using the same graphics, etc. The problem with different architectures between the entity and aggregate simulations would disappear since the same architecture would be used for both simulations. Also, as the network implementation used by the simulation community evolved (HLA, etc), there would be minimal overhead costs involved in keeping the simulation functioning, if any, since the aggregate implementation itself would be ModSAF based. The implementation of the ModSAF aggregate would include tools to automatically generate the data files that determine the aggregate's interaction results. This would allow automatic recalibration of aggregate data files if or when changes are made to the entity ModSAF or the data files upon which previous calibrations were based. This would significantly reduce overhead and maintenance costs as the ModSAF entity simulation evolves, and allow individual experimenters making changes to entity ModSAF to easily alter their own copy of the aggregate simulation to reflect resulting changes in interaction dynamics. Also, since this aggregate simulation would be ModSAF based, and since it would be calibrated with the ModSAF entity scenarios, it would have the best chance of any aggregate simulation of accurately predicting or reproducing the probable outcome of any ModSAF entity scenario or portion of a scenario it was used to replace. This would give similar results from a scenario regardless of whether it was run with or without aggregates.

Many of the implementations used in this experiment are the first step in creating such a ModSAF aggregate simulation. The current simple aggregate would act as a baseline or model for a more encompassing aggregate implementation. The Lanchester equations could be expanded to cover a varied range of unit types. The resulting equations and their outcomes could be examined to determine if there needs to be a more complex or descriptive implementation to better reflect the underlying entity interactions, involving other variables such as distance between units, and possibly terrain analysis. The tools created to aid in the calibration for this experiment could be automated, and scenarios added to make the calibration of the new parameters to the Lanchester equations automatic. This would make calibration and use of the aggregate simulation almost transparent to the experimenter in the field with their own copy of ModSAF.

References

Davis, Paul K. (1995) "Aggregation, Disaggregation, and the 3:1 Rule in Ground Combat" *ElecSim '95 Electronic Conference on Scalability in Training Simulation*, April 10th - June 18th.

Davis, Paul, K. and Hillestad, Richard (1993) "Families of Models that Cross Levels of Resolution: Issues for Design, Calibration and Management" Proceeding of the 1993 Winter Simulation Conference, December, Los Angeles, California.

Davis, Paul, K. *An Introduction to Variable-Resolution Modeling and Cross-Resolution Model Connection* RAND R-4242-DARPA.

Franceschini, Robert W. and Clark, Clark R. (1994) " Integrated Eagle/BDS-D: Results and Current Work" *Proceedings of the 11th Workshop on Standards for the Interoperability of Distributed Simulations* September 26-30, pp. 419-423.

Fricker, Ronald D. (1996) "Lanchester Models of the Ardennes Campaign" Yale University, Published on the Internet.

Hillestad, Richard J. and Juncosa, Mario L. *Cutting Some Trees to See the Forest: On Aggregation and Disaggregation in Combat Models* (RAND R-4250-DARPA).

Hillestad, Richard J., Owen, John and Blumenthal, Don *Experiments in Variable-Resolution Combat Modeling* (RAND N-3631-DARPA).

Kaminski, Paul G. (1996) "DoD High Level Architecture (HLA) for Simulation (M&S) Memorandum" DMSO News Vol. 1, No. 3, Fall.

Karr, Clark R. and Root, Eric (1994) "Integrating Aggregate and Vehicle Level Simulations" *Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation* July 23-25, Orlando, Florida, pp. 425-435.

Kester, James E. (1996) "An Approach to Aggregation and Deaggregation for J-Mass System Models" *15th Workshop on the Interoperability of Distributed Interactive Simulation September 16-29, 1996* Simulation Training and Instrumentation Command Defense Modeling and Simulation Office, University of Central Florida, Institute for Simulation & Training.

Lanchester, F. W. (1916) *Aircraft in Warfare: The Dawn of the Fourth Arm* Constable, London.

Massey, Dan (1995) *Feasibility Analysis Report* Report No. ADSTII-CDRL-A009-002.

Natrajan, Anand and Nguyen-Tuong, Anh (1995) "To Disaggregate or Not To Disaggregate, That is *Not* The Question" *ElecSim '95 Electronic Conference on Scalability in Training Simulation*, April 10th - June 18th.

Peacock, Jeffrey C., Bombardier, Kevin C.and Panagos, James (1996) "The JPDS Corps Level Computer Generated Forces (CLCGF) System Project Update 1996" *Proceedings of the Sixth Conference on Computer Generated Forces and Behavioral Representation* July 23-25, Orlando, Florida, pp. 291-301.

Pratt, David R. and Johnson, Matthew A. (1995) "Constructive and Virtual Model Linkage" Proceeding of the 1995 Winter Simulation Conference, Society for Computer Simulation.

Schricker, Stephen A., Franceschini, Robert W., Stober, David R. and Nida, Jonathan C. (1996) "An Architecture for Linking Aggregate and Virtual Simulations" *Proceedings of the Sixth Conference on Computer Generated Forces and Behavioral Representation* July 23-25, Orlando, Florida, pp. 427-434.

Seidel, David, Salisbury, Marnie R., Booker, Lashon B., and Dahmann, Judith (1995). "CFOR Approach to Simulation Scalability" *ElecSim '95 Electronic Conference on Scalability in Training Simulation*, April 10th - June 18th.

Stober, David, Kraus, Matthew K., Foss, William F., Franceschini, Robert W., and Petty, Mikel D. (1995) "Survey of Constructive + Virtual Linkages" *Proceedings of the Fifth Conference on*

Computer Generated Forces and Behavioral Representation May 9-11, Orlando, Florida, pp. 93-102.

Wise, Ben (1991) “Lanchester Equation for Automated Planning” *Phalanx* March,pp. 24-36.

Appendix A: Software Requirements and Design

The requirements for this contract were that results be obtained for Virtual and Constructive engagements, and compared. As such, the results were the main focus and deliverable of the development, not the software. With this in mind, the first implementation of the Constructive ModSAF and the data recording and time saving tools were done as quick and dirty implementations, so that if something unexpected happened that set us behind schedule, we would have been at minimal risk for not completing what was required. Only after the preliminary results were obtained were these implementations rewritten to be in a more clean and manageable implementation that did not interfere with normal operation of ModSAF, and that could potentially be reused. The requirements also did not specify that software be written to perform the mathematics involved in much of the data parameter refinement process, but it was felt that by doing so there was far less chance for the possibility of errors in the process, and in the long run it would be less actual work. Again, this code also could potentially be reusable, and if there were future work on this project, this software could help in possibly automating the process.

Appendix B: Input Parameters for ModSAF (C)

The following list input parameters and input number-of-tanks values/output number-of-surviving-tanks values for each of the runs made for the calibration of Lanchester parameters for ModSAF (C).

Input Values/Data Results for ModSAF (C), Adjustment Run 1

The following are input parameters to ModSAF(C):

- engagement_range 3500.0
- defense_factor 0.602588
- assault_factor 1.0
- red_tank_per_co 10 (changes with the factor: Force Level)
- blue_tank_per_co 10 (changes with the factor: Force Level)
- attrition_factor_red 0.023354 (rate red takes damage from number blue tanks)
- attrition_factor_blue 0.12865 (rate blue takes damage from number red tanks)

Table with output results from Run 1:

Friendly Start	Enemy Start	Friendly End	Enemy End
10	30	0.0	29.45
14	30	0.0	28.92
18	30	0.0	28.18
14	42	0.0	41.23
30	10	2.17	3.67
42	10	27.70	0.0
54	10	43.84	0.0
42	14	10.37	5.66

Input Values/Data Results for ModSAF (C), Adjustment Run 2

The following are input parameters to ModSAF(C):

- engagement_range 3500.0
- defense_factor 0.5346
- assault_factor 1.0
- red_tank_per_co 10 (changes with the factor: Force Level)
- blue_tank_per_co 10 (changes with the factor: Force Level)

- attrition_factor_red 0.0189438 (rate red takes damage from number blue tanks)
- attrition_factor_blue 0.117294 (rate blue takes damage from number red tanks)

Table with output results from Run 2:

Friendly Start	Enemy Start	Friendly End	Enemy End
10	30	0.0	29.57
14	30	0.0	29.15
18	30	0.0	28.57
14	42	0.0	41.40
30	10	4.32	3.95
42	10	27.68	0.0
54	10	43.86	0.0
42	14	7.31	5.65

Input Values/Data Results for ModSAF (C), Adjustment Run 3

The following are input parameters to ModSAF(C):

- engagement_range 3500.0
- defense_factor 0.532223
- assault_factor 1.0
- red_tank_per_co 10 (changes with the factor: Force Level)
- blue_tank_per_co 10 (changes with the factor: Force Level)
- attrition_factor_red 0.01923 (rate red takes damage from number blue tanks)
- attrition_factor_blue 0.118119 (rate blue takes damage from number red tanks)

Table with output results from Run 3:

Friendly Start	Enemy Start	Friendly End	Enemy End
10	30	0.0	29.57
14	30	0.0	29.15
18	30	0.0	28.57
14	42	0.0	41.40
30	10	4.91	3.86

42	10	27.92	0.0
54	10	43.99	0.0
42	14	8.09	5.50

Appendix C: Input Values and Simulation Results

The following table contains data generated from the ModSAF (V) runs and runs made from the calibrated ModSAF (C). The observations recorded are number of tanks input to the execution and number of tanks the are battle capable at the end of the run.

Posture	Friendly Start	Enemy Start	Entity Friendly End	Entity Enemy End	Aggregate Friendly End	Aggregate Enemy End
F-attack	10	30	0	30	0	29.57
F-attack	10	30	0	30	0	29.57
F-attack	10	30	0	30	0	29.57
F-attack	10	30	0	30	0	29.57
F-attack	10	30	0	29	0	29.57
F-attack	10	30	0	29	0	29.57
F-attack	10	30	0	29	0	29.57
F-attack	10	30	0	30	0	29.57
F-attack	10	30	0	29	0	29.57
F-attack	10	30	0	30	0	29.57
F-attack	10	30	0	30	0	29.57
F-attack	10	30	0	30	0	29.57
F-attack	10	30	0	30	0	29.57
F-attack	10	30	0	30	0	29.57
F-attack	10	30	0	30	0	29.57
F-attack	10	30	0	30	0	29.57
F-attack	10	30	0	30	0	29.57
F-attack	10	30	0	30	0	29.57
F-attack	10	30	0	30	0	29.57
F-attack	10	30	0	30	0	29.57
F-attack	10	30	0	30	0	29.57
F-attack	10	30	0	30	0	29.57
F-attack	10	30	0	30	0	29.57
F-attack	10	30	0	29	0	29.57
F-attack	10	30	0	29	0	29.57
F-attack	10	30	0	30	0	29.57
F-attack	10	30	0	30	0	29.57
F-attack	10	30	0	30	0	29.57
F-attack	10	30	0	30	0	29.57
F-attack	10	30	0	30	0	29.57
F-attack	18	30	0	28	0	28.57
F-attack	18	30	0	26	0	28.57
F-attack	18	30	0	30	0	28.57
F-attack	18	30	0	30	0	28.57
F-attack	18	30	0	30	0	28.57
F-attack	18	30	0	30	0	28.57
F-attack	18	30	0	29	0	28.57
F-attack	18	30	0	29	0	28.57
F-attack	18	30	0	29	0	28.57
F-attack	18	30	0	28	0	28.57

F-attack	18	30	0	30	0	28.57
F-attack	18	30	0	29	0	28.57
F-attack	18	30	0	30	0	28.57
F-attack	18	30	0	24	0	28.57
F-attack	18	30	0	25	0	28.57
F-attack	18	30	0	28	0	28.57
F-attack	18	30	0	30	0	28.57
F-attack	18	30	0	30	0	28.57
F-attack	18	30	0	28	0	28.57
F-attack	18	30	0	30	0	28.57
F-attack	18	30	0	30	0	28.57
F-attack	18	30	0	30	0	28.57
F-attack	18	30	0	28	0	28.57
F-attack	18	30	0	28	0	28.57
F-attack	18	30	0	28	0	28.57
F-attack	18	30	0	27	0	28.57
F-attack	18	30	0	26	0	28.57
F-attack	18	30	0	29	0	28.57
F-attack	18	30	0	28	0	28.57
F-attack	18	30	0	26	0	28.57
F-attack	14	42	0	41	0	41.4
F-attack	14	42	0	40	0	41.4
F-attack	14	42	0	40	0	41.4
F-attack	14	42	0	41	0	41.4
F-attack	14	42	0	39	0	41.4
F-attack	14	42	0	39	0	41.4
F-attack	14	42	0	42	0	41.4
F-attack	14	42	0	40	0	41.4
F-attack	14	42	0	42	0	41.4
F-attack	14	42	0	41	0	41.4
F-attack	14	42	0	41	0	41.4
F-attack	14	42	0	40	0	41.4
F-attack	14	42	0	38	0	41.4
F-attack	14	42	0	41	0	41.4
F-attack	14	42	0	41	0	41.4
F-attack	14	42	0	41	0	41.4
F-attack	14	42	0	41	0	41.4
F-attack	14	42	0	41	0	41.4
F-attack	14	42	0	39	0	41.4
F-attack	14	42	0	41	0	41.4
F-attack	14	42	0	36	0	41.4
F-attack	14	42	0	39	0	41.4
F-attack	14	42	0	41	0	41.4
F-attack	14	42	0	39	0	41.4
F-attack	14	42	0	39	0	41.4
F-attack	14	42	0	39	0	41.4
F-attack	14	42	0	39	0	41.4
F-attack	14	42	0	40	0	41.4
F-attack	14	42	0	39	0	41.4

F-attack	14	42	0	40	0	41.4
F-attack	14	30	0	30	0	29.15
F-attack	14	30	0	28	0	29.15
F-attack	14	30	0	29	0	29.15
F-attack	14	30	0	28	0	29.15
F-attack	14	30	0	28	0	29.15
F-attack	14	30	0	30	0	29.15
F-attack	14	30	0	29	0	29.15
F-attack	14	30	0	29	0	29.15
F-attack	14	30	0	30	0	29.15
F-attack	14	30	0	30	0	29.15
F-attack	14	30	0	28	0	29.15
F-attack	14	30	0	30	0	29.15
F-attack	14	30	0	30	0	29.15
F-attack	14	30	0	30	0	29.15
F-attack	14	30	0	30	0	29.15
F-attack	14	30	0	29	0	29.15
F-attack	14	30	0	28	0	29.15
F-attack	14	30	0	28	0	29.15
F-attack	14	30	0	30	0	29.15
F-attack	14	30	0	29	0	29.15
F-attack	14	30	0	30	0	29.15
F-attack	14	30	0	29	0	29.15
F-attack	14	30	0	30	0	29.15
F-attack	14	30	0	30	0	29.15
F-attack	14	30	0	28	0	29.15
F-attack	14	30	0	29	0	29.15
F-attack	14	30	0	30	0	29.15
F-attack	14	30	0	29	0	29.15
F-attack	14	30	0	28	0	29.15
F-attack	14	30	0	29	0	29.15
E-attack	30	10	8	1	4.91	3.86
E-attack	30	10	5	3	4.91	3.86
E-attack	30	10	14	0	4.91	3.86
E-attack	30	10	7	1	4.91	3.86
E-attack	30	10	8	1	4.91	3.86
E-attack	30	10	7	5	4.91	3.86
E-attack	30	10	3	3	4.91	3.86
E-attack	30	10	16	0	4.91	3.86
E-attack	30	10	10	0	4.91	3.86
E-attack	30	10	10	5	4.91	3.86
E-attack	30	10	6	3	4.91	3.86
E-attack	30	10	5	3	4.91	3.86
E-attack	30	10	8	1	4.91	3.86
E-attack	30	10	14	3	4.91	3.86
E-attack	30	10	16	0	4.91	3.86
E-attack	30	10	11	5	4.91	3.86
E-attack	30	10	10	1	4.91	3.86
E-attack	30	10	14	0	4.91	3.86

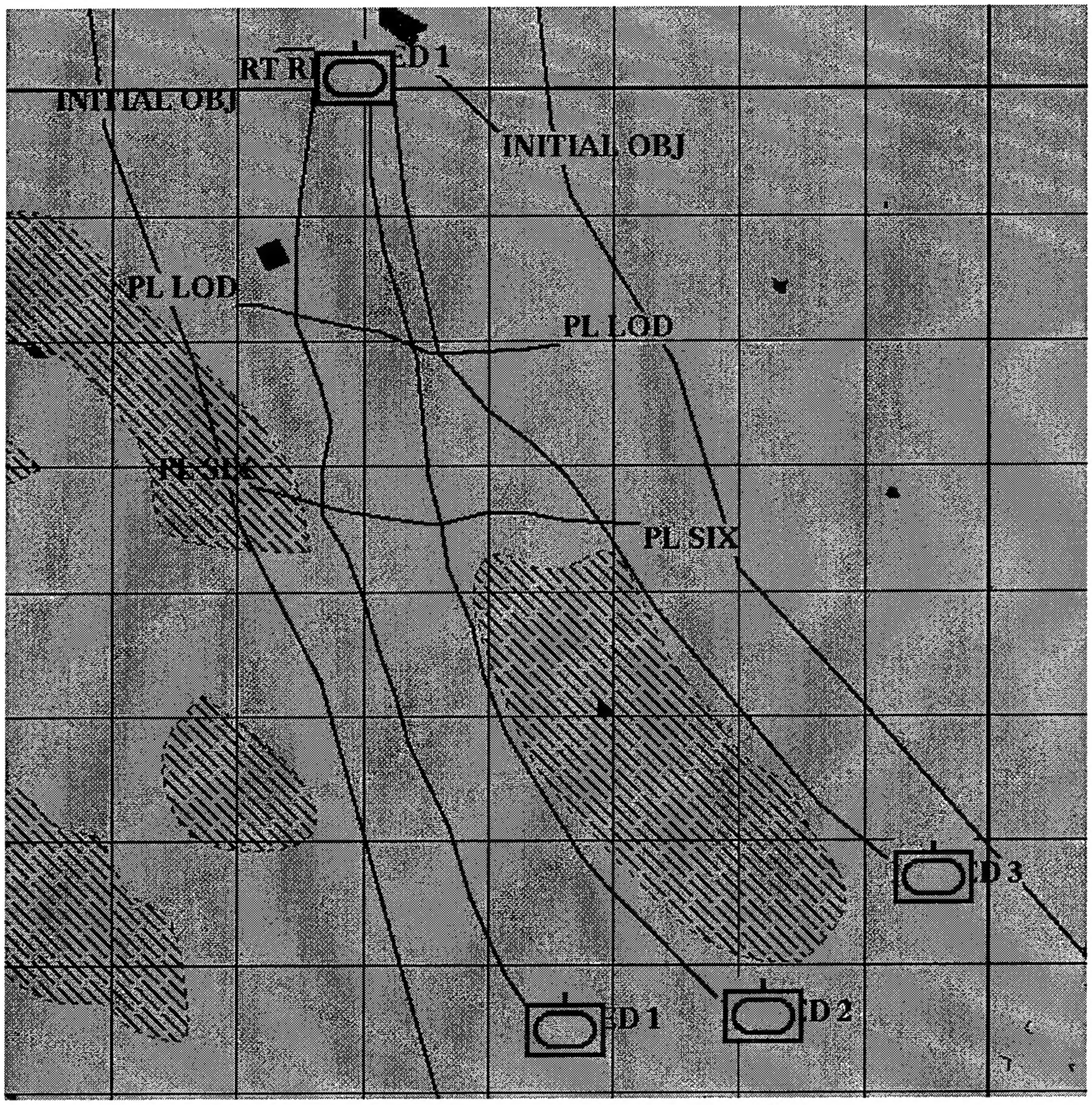
E-attack	30	10	16	1	4.91	3.86
E-attack	30	10	12	0	4.91	3.86
E-attack	30	10	4	3	4.91	3.86
E-attack	30	10	8	1	4.91	3.86
E-attack	30	10	18	1	4.91	3.86
E-attack	30	10	21	3	4.91	3.86
E-attack	30	10	13	2	4.91	3.86
E-attack	30	10	8	6	4.91	3.86
E-attack	30	10	19	0	4.91	3.86
E-attack	30	10	8	3	4.91	3.86
E-attack	30	10	10	1	4.91	3.86
E-attack	30	10	14	0	4.91	3.86
E-attack	54	10	30	2	43.99	0
E-attack	54	10	41	1	43.99	0
E-attack	54	10	37	0	43.99	0
E-attack	54	10	37	1	43.99	0
E-attack	54	10	37	0	43.99	0
E-attack	54	10	38	0	43.99	0
E-attack	54	10	40	0	43.99	0
E-attack	54	10	37	0	43.99	0
E-attack	54	10	32	0	43.99	0
E-attack	54	10	37	0	43.99	0
E-attack	54	10	32	0	43.99	0
E-attack	54	10	41	1	43.99	0
E-attack	54	10	40	0	43.99	0
E-attack	54	10	34	1	43.99	0
E-attack	54	10	40	0	43.99	0
E-attack	54	10	36	0	43.99	0
E-attack	54	10	36	0	43.99	0
E-attack	54	10	32	0	43.99	0
E-attack	54	10	36	0	43.99	0
E-attack	54	10	37	0	43.99	0
E-attack	54	10	44	1	43.99	0
E-attack	54	10	35	1	43.99	0
E-attack	54	10	45	0	43.99	0
E-attack	54	10	39	0	43.99	0
E-attack	54	10	30	0	43.99	0
E-attack	54	10	35	0	43.99	0
E-attack	54	10	30	1	43.99	0
E-attack	54	10	38	0	43.99	0
E-attack	54	10	34	0	43.99	0
E-attack	54	10	42	0	43.99	0
E-attack	42	14	25	0	8.09	5.5
E-attack	42	14	16	1	8.09	5.5
E-attack	42	14	10	2	8.09	5.5
E-attack	42	14	17	0	8.09	5.5
E-attack	42	14	13	3	8.09	5.5
E-attack	42	14	15	3	8.09	5.5
E-attack	42	14	20	3	8.09	5.5

E-attack	42	14	19	6	8.09	5.5
E-attack	42	14	19	7	8.09	5.5
E-attack	42	14	14	0	8.09	5.5
E-attack	42	14	17	0	8.09	5.5
E-attack	42	14	15	0	8.09	5.5
E-attack	42	14	12	1	8.09	5.5
E-attack	42	14	15	0	8.09	5.5
E-attack	42	14	13	7	8.09	5.5
E-attack	42	14	10	4	8.09	5.5
E-attack	42	14	18	0	8.09	5.5
E-attack	42	14	9	5	8.09	5.5
E-attack	42	14	21	0	8.09	5.5
E-attack	42	14	8	0	8.09	5.5
E-attack	42	14	22	1	8.09	5.5
E-attack	42	14	18	2	8.09	5.5
E-attack	42	14	9	3	8.09	5.5
E-attack	42	14	15	8	8.09	5.5
E-attack	42	14	22	4	8.09	5.5
E-attack	42	14	11	4	8.09	5.5
E-attack	42	14	24	0	8.09	5.5
E-attack	42	14	16	0	8.09	5.5
E-attack	42	14	15	0	8.09	5.5
E-attack	42	14	16	3	8.09	5.5
E-attack	42	10	25	2	27.92	0
E-attack	42	10	26	0	27.92	0
E-attack	42	10	24	2	27.92	0
E-attack	42	10	27	0	27.92	0
E-attack	42	10	23	0	27.92	0
E-attack	42	10	26	0	27.92	0
E-attack	42	10	24	2	27.92	0
E-attack	42	10	23	3	27.92	0
E-attack	42	10	27	1	27.92	0
E-attack	42	10	26	0	27.92	0
E-attack	42	10	23	1	27.92	0
E-attack	42	10	24	0	27.92	0
E-attack	42	10	28	0	27.92	0
E-attack	42	10	29	0	27.92	0
E-attack	42	10	29	1	27.92	0
E-attack	42	10	29	0	27.92	0
E-attack	42	10	21	0	27.92	0
E-attack	42	10	31	0	27.92	0
E-attack	42	10	24	0	27.92	0
E-attack	42	10	20	0	27.92	0
E-attack	42	10	26	1	27.92	0
E-attack	42	10	21	1	27.92	0
E-attack	42	10	23	0	27.92	0
E-attack	42	10	25	3	27.92	0
E-attack	42	10	18	1	27.92	0
E-attack	42	10	27	0	27.92	0

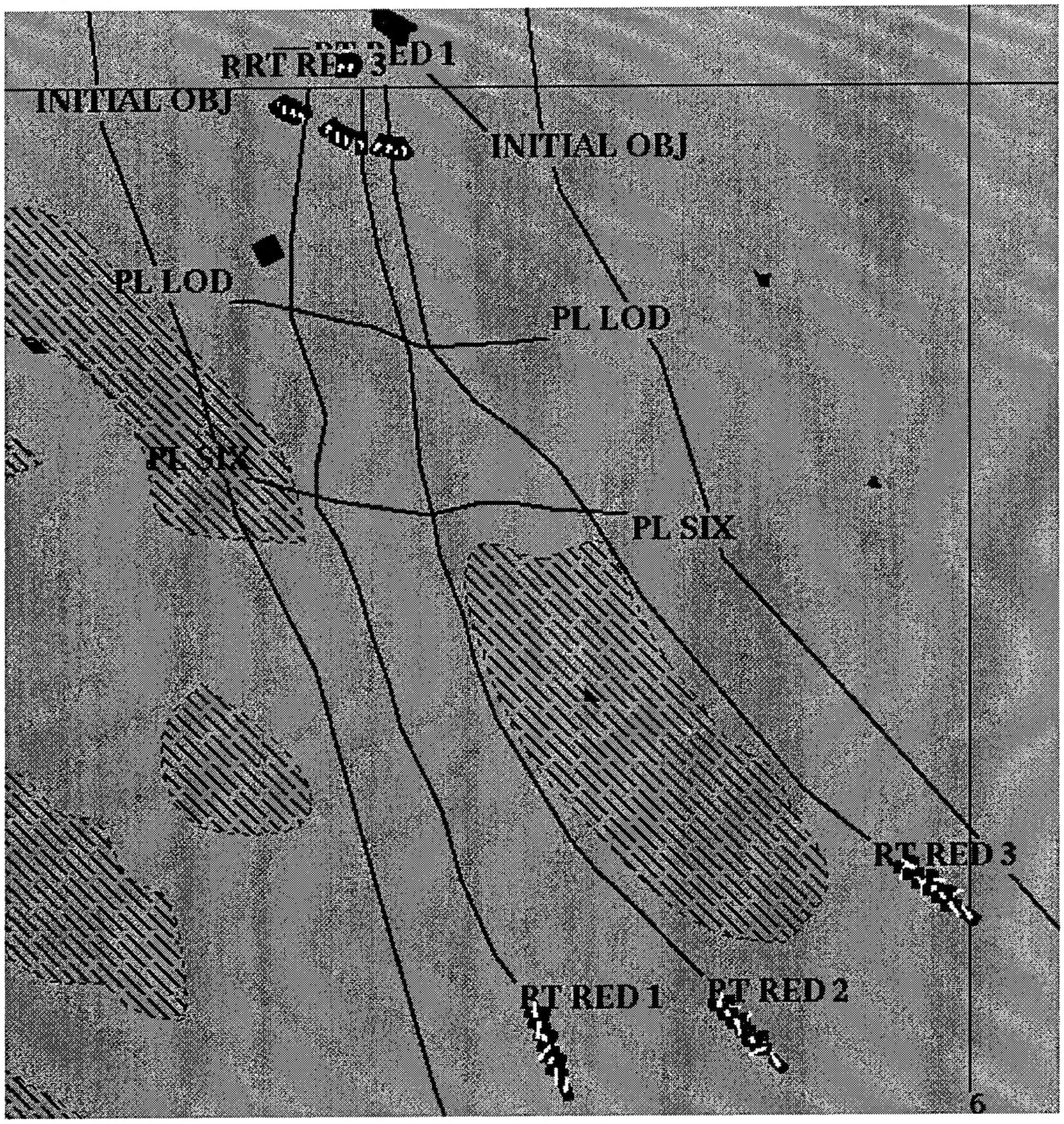
E-attack	42	10	23	0	27.92	0
E-attack	42	10	29	0	27.92	0
E-attack	42	10	30	0	27.92	0
E-attack	42	10	19	0	27.92	0

Appendix D: Screen Prints for ModSAF (V) & (C)

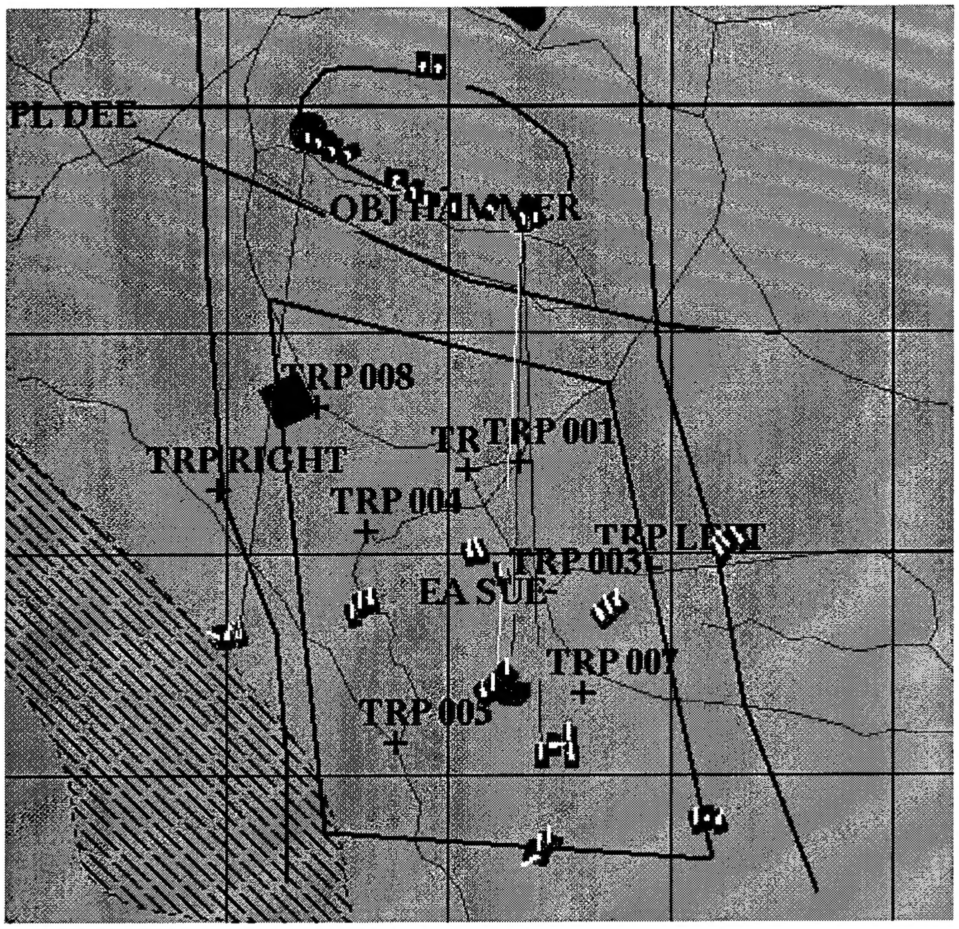
Screen Print from ModSAF (C): start of aggregate level scenario



Screen Print from ModSAF (V): start of entity level scenario



Screen Print from ModSAF (V): entity level combat


PL DEE
OBJ HAMMER
TRP 008
TRP RIGHT
TRP 001
TRP 004
TRP LEFT
TRP 003
EA SUE
TRP 007
TRP 005